CrossTalk

July 2000 The Journal of Defense Software Engineering Vol. 13 No. 7

CMMI

Process Alchemy

CMMI

On the Cover:
Cover artist Tony Peters of L-3 Communications in Layton, Utah mixed the right solution for our first color cover. See p. 5 for more on **CrossTalk**'s new look.

Open Forum

Software Engineering Technology

Departments

CrossTalk

Sponsor	H. Bruce Allgood
Publisher	Reuel S. Alder
Associate Publisher	Lynn Silver
Managing Editor	Kathy Gurchiek
Associate Editor/Layout	Matthew Welker
Associate Editor/Features	Heather Winward
Voice	801-775-5555
Fax	801-777-8069
E-mail	crosstalk.staff@hill.af.mil
STSC Online	www.stsc.hill.af.mil
CrossTalk Online	www.stsc.hill.af.mil/Crosstalk/crostalk.html
CRSIP Online	www.crsip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may use the form on page 31.

Ogden ALC/TISE
7278 Fourth Street
Hill AFB, Utah 84056-5205

Impacting the Future of Process Improvement

This issue of CROSSTALK is focused on introducing the Capability Maturity Model Integration (CMMI) project. The Computer Resources Support Improvement Program (CRSIP) office dedicated a significant amount of our resources to the project over the past two years. The former CRSIP Director, Lt. Col. (Ret.) Joe Jarzombek and I each dedicated at least one week per month to its development. We were assigned to the Requirements and Training Integrated Product Teams (IPT), respectively. We participated as authors on the Product Development Team. A reprint of our interview talking about Measurement and Analysis, *Up Close with Lt.Col. (Ret.) Joe Jarzombek and Bruce Allgood,* is on page 4. Our office also sponsors additional Air Force personnel to participate in other IPTs to assist in developing this product suite. This commitment is a reflection of our belief in the value of CMMI's ability to impact the future of process improvement as a true enterprise-wide effort.

A good place to start to understand CMMI is with the article authored by Joan Weszka, Phil Babel, and Jack Ferguson, *CMMI: An Evolutionary Path to Enterprise Process Improvement* on page 8. Joan served on the Steering Group for this project with Phil, who was the co-chairman prior to his retirement from the Air Force. Dr. Jack Ferguson was the project manager for the first 20 months of the project prior to accepting his current position in the Office of the Secretary of Defense. Joan (industry), Phil (government), and Jack (academia) represent the three collaborative groups brought together to accomplish this effort. Their article tells the history, motivation, drivers, and sponsors of the collaborative effort as well as how the project and organization of the new model was organized. Anticipated benefits, release plans and transition plans are covered for readers.

Ronald Starbuck's article, *A Configuration Manager's Perspective* on page 12 provides a nondeveloper point of view about how this new model differs in the treatment of one particularly significant process area—configuration management. He comments on some of the more notable changes in the new model as compared to the SW-CMM Version 1.1 release that has wide acceptance and use.

Of additional use is the article by Aaron Clouse and Curt Wells, *Transitioning from EIA/IS-731 to CMMI* found on page 15. They outline what this model means to users of the EIA 731 interim standard that the International Council of Systems Engineering developed. These two Product Development Team members give a good explanation of what is required by systems engineers familiar with EIA 731 to understand and consider as they plan transitioning to CMMI usage.

One notable facet of the evolution of CMMs, and how CMMI will assist the engineering community in transitioning to true enterprise-wide process improvement, is the built-in extensibility of CMMI. This effort was not simply a combination of the existing models to create a single new model. Rather, the project was the creation of a framework in which additional extensions of areas of interest can be added to the model in a straightforward, consistent manner allowing for minimization of overlap and maximum reusability of basic model content. Areas of particular interest being proposed at this time for extension and addition to CMMI are Systems Acquisition and Systems Safety. Proponents for these additional process areas have started the steps necessary to integrate new content into the CMMI framework by proposing new process areas covering these new specific disciplines.

The Air Force CRSIP office will continue to support and champion the development and transition of CMMI as an Enterprise-Wide Process Improvement Tool. I invite you to study these articles, as well as to look for further CMMI-related articles in future issues of CROSSTALK.

H. Bruce Allgood
Deputy CRSIP Director

Up Close with Lt. Col. (Ret.) Joe Jarzombek and Bruce Allgood

Lt. Col. Joe Jarzombek retired in April as the Director of CRSIP. He managed CRSIP efforts in the STSC to provide technology information services, such as *CROSSTALK: the Journal of Defense Software Engineering*. In addition to the CMMI Product Development Team, he has served on various software steering committees and working groups to provide coordinated software technology initiatives, policies, and practices.

H. Bruce Allgood is the Deputy of the Computer Resources Support Improvement Program (CRSIP) at Hill AFB in Ogden, Utah. He brings 29 years of experience in the electronics and software marketplace to his assignment. He has been involved with the development of Practical Software Measurement (PSM) since its inception in 1994. Allgood has been involved with software process improvement efforts utilizing the Capability Maturity Model (CMM®) for the past eight years. As a member of the Air Force Software Technology Support Center (STSC), he supports software process improvement efforts throughout the Air Force and DoD. He has worked with Capability Maturity Model Integration (CMMI), assigned to work with the Training Integrated Product Team (IPT) within the CMMI Product Development Team. He represents the Air Force and DoD perspective to the CMMI training effort.

The importance of measurement and its use in the field of software process improvement have been steadily increasing in recent years. To shed more light on the growing use of this discipline, Scott Lucero, Program Manager of the Army Software Metrics Office, interviewed H. Bruce Allgood and Lt. Col. Joe Jarzombek on the Capability Maturity Model®-Integrated-Systems/Software Engineering (CMMI-S/SE)'s new process area for measurement.

Measurement has been elevated to the status of a separate process area in the CMMI that Allgood and Lt. Col. Jarzombek helped author. Lucero, who has been published in CROSSTALK, interviewed the men on the CMMI's new process area for measurement.

The following interview is reprinted courtesy of *Insight, The Army's Software Metrics Newsletter.* It first appeared in the Spring 1999 issue (Vol. 3, No. 4)

The Interview

Question: The first public release draft of the CMMI stated that there are more than 30 process improvement models that use the general approach of the CMM. These models include the Software CMM, the Systems Engineering CMM, and the People CMM. What is the relationship between the CMMI and the other CMM-based models?
Answer: The CMMI effort uses three documents as source models:

- CMM for Software, Version 2, draft C.
- Electronic Industries Alliance Interim Standard (EIA/IS) 731, System Engineering Capability Model (SECM).
- Integrated Product Development Capability Maturity Model, draft Version 0.98 (IPD-CMM).

Several additional CMM-related documents are listed as references for the effort, including the Software Acquisition CMM (SA-CMM) and the Federal Aviation Administration Integrated CMM (iCMM).

Question: Measurement is already an integral part of several Key Process Areas (KPAs) of the CMM. Why create a separate process area for measurement in the CMMI? Is it due, in part, to the Level 4 quantitatively managed stage of the CMM?
Answer: One of the requirements found in the CMMI A-Specification is to have the product suite consistent and compatible with the International Standards Organization/International Electrotechnical Commission (ISO/IEC) 15504, which includes a measurement process area. It was felt that, even though measurement is referenced in several of the CMM KPAs, there was insufficient focus on measurement at the lower maturity levels. Organizations that have achieved the highest CMM ratings have reported to us that a clear focus on measurement at lower levels would have saved them significant efforts later. Yes, the need for a measurement process area in CMMI is due in part to the requirements for Level 4. Historical measurement data is necessary to be able to quantitatively manage processes. It was also felt that without measurement as a process area, management would not pay adequate attention to measurement as a critical success factor for process improvement.

Question: Can you tell us a little about the Measurement Process Area (PA)? How is it different from the measurement aspects of the different KPAs of the Software CMM?
Answer: Although several existing process areas have measurement and analysis as common features, without the Measurement and Analysis PA there is no one place in the CMM where practitioners can go to find a description of good measurement practice. The Measurement and Analysis PA provides a focus area and foundation for the various applications of measurement in project management and process improvement activities. The PA provides greater consistency and understanding with respect to the practice of measurement. Therefore, the CMMI should allow organizations to implement measurement more easily than if the equivalent practices were spread across multiple process areas. We are afraid that management buy-in to the need for good measurement processes will be missing if measurement is not raised to a process area.

Question: Could you briefly describe the two representations of the CMMI, continuous and staged, and the need for these two different representations?

Answer: The CMMI A-Specification explicitly requires both a staged representation and a continuous representation.[1] In a staged representation, such as the SW-CMM, each maturity level contains a specific set of process areas that must be achieved before moving to a higher maturity level. The continuous representation, such as the SECM, has only a recommended sequence of process areas that should be achieved. To accommodate this variation of source models, the CMMI product suite offers staged and continuous representations of each CMMI model. Some parts of an organization may prefer the staged representation, while others may prefer the continuous representation. The content of each model representation is virtually identical. Regardless of which representation is used, assessments using either should produce very similar results, and the guidance stemming from an assessment should be the same.

Question: The CMMI is based on the FAA's Integrated Capability Maturity Model (iCMM) work,[2] which integrates the Software and Systems Engineering CMMs. Was measurement a part of the FAA-iCMM?
Answer: Yes, measurement is a process area in FAA-iCMM. Authors of the CMMI Measurement and Analysis PA used the information created by FAA-iCMM as a reference.

Question: Was not the Software Technology Support Center (STSC) involved in the trial applications of the FAA-iCMM in the Department of Defense? What has been the experience in the pilot applications of the iCMM?
Answer: Yes, the STSC is involved in piloting enterprise-wide process improvement efforts at Warner Robins Air Logistic Center using the FAA-iCMM as a model. Using the same reference model has proven useful in getting the software, systems, and acquisition communities to work closely in achieving common goals.[3]

Question: Einstein said that our theories determine what we measure. The SEI has had various takes on software measurement over the years: the SEI core measures, the Goal-Question-Indicator-Metric, and the CMM measurement map. Does the new Measurement PA, in essence, create a new theory about what aspects of software development to measure?
Answer: There is nothing in the CMMI Measurement and Analysis PA that is considered new theory. The PA is based on industry-best practices for measurement and does not dictate specific measures, or presuppose how measurement must be accomplished for a particular project.

Question: There is an effort to create an international standard for software measurement, ISO 15939, which is partially based on the Practical Software Measurement (PSM) guidance. What is the relationship between the CMMI Measurement PA and ISO 15939? Are these two efforts in sync?
Answer: Authors of the CMMI Measurement PA actually used the ISO 15939 document as a resource to create the Measurement and Analysis PA. As a result, the philosophy found in the process area closely follows that found in PSM. Efforts are definitely in sync and are expected to remain so.

Question: Lastly, why are you all known as the *Measurement Mafia*?
Answer: Initial voting on what process areas to include in the CMMI resulted in a split vote on measurement. A few of the Product Development Team (PDT) members, the most vociferous, were given the task to create the new process area for further study and consideration. After this group of PDT members created measurement and analysis as a new process area, it lobbied other members with the reasons why it ought to be accepted as a PA. They soon became known as the *Measurement Mafia.* I assume this was a light-hearted reflection on our persuasive tactics. ✍

Notes

1. See page 6 for more on this in an article by Sandy Shrum.
2. See page 8 for more on this in an article by Joan Weszka, Phil Babel, and Jack Ferguson.
3. A report on the progress of this effort was presented at STC '99.

A New Look

Readers will notice a new face to CROSSTALK beginning with this issue, with the introduction of color covers. It is the latest in the staff's efforts in the past year to bring you a quality product, with such improvements as:

- The F-22 poster insert and quiz in the May issue.
- A more user-friendly and better positioned Table of Contents.
- A cleaner, sharper layout to make the contents more attractive and more readable.
- The addition of our Web site address on the bottom of the pages throughout the journal, to make it easier for readers to contact us or access us online.
- Including high-profile interviews with such subjects as Gen. Lester Lyles, former Vice Chief of Staff for the Air Force and now Commander of the Air Force Materiel Command at Wright-Patterson Air Force Base; Dr. Delores Etter, Under Secretary of Defense (S&T); and an upcoming interview with Paul Maritz of Microsoft.
- Listing of theme-related Web sites.

CROSSTALK staff, clockwise from left. *Managing Editor Kathy Gurchiek, Assoc. Editor Heather Winward, Publisher Rudy Alder, Assoc. Editor Matt Welker, Assoc. Publisher Lynn Silver.*

In making these improvements, it was important that we retain those elements that readers have long enjoyed and found useful in CROSSTALK, such as well-written articles on software metrics, process improvement, lessons learned/field reports, and software project management, as well as BACKTALK. We will continue to implement changes—major or minute—that will see CROSSTALK continue as the Department of Defense's premier software engineering journal. As we make these changes, our mission at shall remain the same: *To encourage the engineering development of software in order to improve the reliability, maintainability, and responsiveness to the United States' warfighting capability and to instruct, inform, and educate readers on up-to-date policy decisions and new software engineering technologies.* Drop us a line; let us know how CROSSTALK can meet your needs.

Choosing a CMMI Model Representation

By Sandy Shrum
Software Engineering Institute

What is a CMMI model representation? The answer requires an explanation of the structure of CMMI models. The basic building blocks in every CMMI model are called process areas. A process area does not describe how an effective process is executed (e.g., entrance and exit criteria, roles of participants, resources). It describes what those using an effective process do (practices) and why they do those things (goals).

This article originally appeared in *SEI Interactive,* December 1999, available online at http://interactive.sei.cmu.edu

In a Capability Maturity Model®, process areas can be organized into one of two representations, a continuous representation or a staged representation. For example, the Electronic Industries Association's Interim Standard 731, Systems Engineering Capability Model (SECM) is a model with a continuous representation. The Software Engineering Institute's Capability Maturity Model for Software (SW-CMM®) is a model with a staged representation.

To illustrate why an organization might choose one representation over the other, imagine two companies, Foo Toys and Widget Toys. Both companies manufacture software-intensive toys and, until now, have not pursued process improvement.

The Foo Toys management wants to improve how the company handles risks and integrates product components. Management is happy with how the company's other processes are operating and decides to focus on those two process areas only. Foo Toys' management chooses the continuous representation. Using that representation, Foo Toys will concentrate on only those process areas that relate to risk management and the integration of components. When Foo Toys achieves both the specific goals for a process area and the general goals associated with all levels equal to or less than a particular capability level, it achieves the capability level for that process area. (Goal achievement is determined by a review of the practices associated with the goal.) If Foo Toys successfully achieves the specific goals for product integration and all the capability Level 2 and 3 goals, it could be said that Foo Toys is Level 3 in product integration.

The management of Widget Toys, however, wants to improve the company's overall development capability and sees many areas requiring attention. Recognizing the many interdependencies across process areas, Widget Toys' management chooses the staged representation. Using that representation, Widget Toys will concentrate on the process areas at maturity Level 2, thus establishing its project management processes. When Widget Toys performs the practices in these process areas successfully, it also achieves the corresponding goals. When Widget Toys achieves all of the goals of a process area, the process area is satisfied. For Widget Toys to successfully achieve a maturity level, it must satisfy all of the process areas through that level. If Widget Toys satisfies all of the process areas through maturity Level 2, it could be said that Widget Toys is maturity Level 2.

By design, the granular information contained in the two CMMI model representations is virtually identical. However, each of the representations provides benefits that will be valued differently by organizations.

In CMMI models, process areas describe key aspects of such processes as configuration management, requirements management, product verification, systems integration, and many others. Let us examine the two representations in more detail.

Continuous Representation

In the continuous representation of a CMMI model, the summary components are process areas. Within each process area there are specific goals that are implemented by specific practices. Also contained in the continuous representation of a CMMI model are generic goals that are implemented by generic practices.

Specific goals and practices are unique to individual process areas, whereas generic goals and practices apply to multiple process areas. Each practice belongs to only one capability level. To satisfy capability Level 2 for a process area, Foo Toys must satisfy the specific goals and Level 2 practices for that process area as well as the Level 2 generic goals for that same process area.

Staged Representation

In the staged representation, the summary components are maturity levels. Within each maturity level there are process areas that contain goals, common features, and practices. For Widget Toys, the practices serve as guides on what to implement to achieve the goals of the process area.

In a staged representation of a CMMI model, practices are categorized into common features:

1. *Commitment to perform* includes practices that ensure that the process is established and will endure. It typically involves establishing organizational policies and leadership.
2. *Ability to perform* includes practices that establish the necessary conditions for implementing the process completely. It typically involves plans, resources, organizational structures, and training.
3. *Activities performed* includes practices that directly implement a process. These practices distinguish a process area from others.
4. *Directing implementation* includes practices that monitor and control the performance of the process. These typically involve placing designated work products of the process under configuration management, monitoring and controlling the performance of the process against the plan, and taking corrective action.
5. *Verifying implementation* includes practices that ensure compliance with the requirements of the process area. These typically involve reviews and audits.

Capability Levels vs. Maturity Levels

The continuous representation consists of capability levels, while the staged representation consists of maturity levels. The main difference between these two types of levels is the representation they belong to and how they are applied:

- *Capability levels,* which belong to a continuous representation, apply to an organization's process-improvement achievement in individual process areas. There are six capability levels, numbered 0 through 5.
- *Maturity levels,* which belong to a staged representation, apply to an organization's overall process-improvement achievement using the model. There are five maturity levels, numbered 1 through 5. Each maturity level comprises a set of goals that, when satisfied, improve processes. Maturity levels are measured by the achievement of the goals that apply to a set of process areas.

Level	Continuous Representation: Capability Levels	Staged Representation: Maturity Levels
Level 0	Not Performed	N/A
Level 1	Performed	Performed
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	Quantitatively Managed	Quantitatively Managed
Level 5	Optimizing	Optimizing

Table 1. *Capability Levels and Maturity Levels*

When Widget Toys uses the staged representation, it will evaluate its progress using the same basis as all other organizations that use the same model with the staged representation. Although Widget Toys can pursue process improvement at any pace it wishes, the basis for evaluating its progress will be exactly the same.

Using the staged representation, Widget Toys can identify the maturity levels through which it can evolve to establish a culture of engineering excellence. Each maturity level forms a necessary foundation on which to build the next level.

Using the continuous representation, Foo Toys can produce a capability level profile (i.e., a list of process areas and their corresponding capability levels). Types of capability level profiles include the following:

- An *achievement profile* represents the current achieved capability level in selected process areas at Foo Toys.
- A *target profile* represents the capability levels that Foo Toys wishes to achieve.

Maintaining capability level profiles throughout the process-improvement life cycle enables the engineering process group at Foo Toys to demonstrate its progress to management as well as guide its process-improvement activities.

A target profile can reflect the unique needs of the organization (called *target staging*) or it can reflect the levels used by the staged representation (called *equivalent staging*). Equivalent staging permits benchmarking of progress among projects, organizations, and other enterprises.

Selecting a Representation

When making the decision about which architectural representation to use for process improvement, Foo Toys and Widget Toys would consider the comparative advantages of each approach as represented in Table 2:

Continuous Representation	Staged Representation
Grants explicit freedom to select the order of improvement that best meets the organization's business objectives and mitigates the organization's areas of risk.	Introduces a sequence of improvements, beginning with basic management practices and progressing through a predefined and proven path of successive levels, each serving as a foundation for the next.
Enables increased visibility into the capability achieved within each individual process area.	Visibility is primarily at the maturity level with limited visibility at the process area level.
Allows the generic practices from higher capability levels to be more evenly and completely applied to all of the process areas.	Generic practices are grouped as institutionalization common features that are applied to all process areas at all maturity levels.
Because capability levels are measured by process area, comparisons across and among organizations can only be made on a process area by process area basis.	Permits easy comparison across and among organizations because process improvement results are summarized as a single maturity-level number.
Reflects a newer approach that does not yet have the data to demonstrate its ties to return on investment.	Builds on a relatively long history of use that includes case studies and data that demonstrate proven return on investment.
Affords an easy comparison of process improvement to ISO 15504 because the organization of process areas is derived from 15504.	Allows comparison to 15504, but the organization of process areas does not correspond to the organization used in 15504

Table 2. *Advantages of Using Each Model Representation*

Foo Toys chose the continuous representation because it wanted to focus improvement efforts in two predefined areas.

Widget Toys chose the staged representation because it wanted a clear path to process improvement that provides an easy comparison to competitors that use the same model. Regardless of which representation you choose for your organization, the CMMI model you choose will help you improve your development processes. In essence, both representations were designed for equivalent use in process improvement and assessments. ✍

About the Author

Sandy Shrum is a member of the CMMI product-development team and has been a senior writer/editor at the Software Engineering Institute (SEI) since 1995. Before joining the SEI, she spent eight years with Legent Corp., where she was a senior information developer, a member of a software-development team, and a member of Legent's Information Technology organization. She has a master of arts degree in professional writing from Carnegie Mellon University and a bachelor of science degree in business administration and marketing from Gannon University.

4500 5th Ave.
Pittsburgh, Pa. 15213-2612
Voice: 412-268-6503
Fax: 412-268-5758
E-mail: sshrum@sei.cmu.edu

CMMI: Evolutionary Path to Enterprise Process Improvement

By Joan Weszka, *Lockheed Martin*
Phil Babel, *Dayton Aerospace Inc.*
Jack Ferguson, *Office of the Secretary of Defense*

Since 1998, a government-industry-Software Engineering Institute (SEI) collaboration has been under way to develop a product suite of models, training, and assessment methodology that support integrated process and product improvement across the enterprise. These products are intended to replace legacy maturity models, including SW-CMM® and Electronic Industries Association Interim Standard (EIA/IS) 731, the Systems Engineering Capability Model (SECM). This article describes the Capability Maturity Model Integration (CMMI) project, including its drivers, sponsors, organization, scope, products, expected benefits from using the CMMI products, and guidance for transition.

The proliferation of process maturity models and recognition of the inefficiency and ineffectiveness of using multiple, stovepiped models and methods for process improvement was the impetus for the CMMI project. With an initial focus on integrating a subset of existing process maturity models for engineering and Integrated Product and Process Development (IPPD), the CMMI products are being designed using a product line approach to facilitate extension to other disciplines. The objective is to provide a single product suite for enterprise-wide process improvement, with an evolutionary growth path for adoption.

Drivers

Demonstrable benefits from using the SW-CMM v1.1 for process improvement since its release in 1993 have spawned the development of a number of capability models. These have included the Systems Engineering CMM, the Systems Engineering Capability Assessment Method (SECAM), EIA/IS-731, SECM, the Software Acquisition CMM, the People CMM, the System Security Engineering CMM, and the FAA-iCMM. These models, developed by a number of different organizations, have overlapping scopes and lack consistency in architecture, terminology, and assessment methodology. As a result, an organization deploying more than one model is faced with unique training for each, and typically a stovepiped process improvement approach focused on the discipline (e.g., software engineering) covered by the scope of the model. The net effect is often a delta cost of *x* each time an additional model is deployed in an organization, where *x* is the cost of deploying a single model. This situation of multiple models, assessment methods, and training deployed in a single organization, at significant cost, was a catalyst for CMMI.

U.S. Air Force	Boeing	Lockheed Martin
U.S. Army	Comarco Systems	Motorola
U.S. Federal Aviation	Computer Sciences Corp.	Northrop Grumman
U.S. National Security Agency	EER Systems	Pacific Bell
U.S. Navy	Ericsson Canada/Q-Labs	Priority Tech
Software Engineering Institute	General Dynamics	Raytheon
ADP, Inc.	Harris Corp.	Rockwell Collins
Andersen Consulting	Honeywell	Sverdrup Corp.
AT&T Labs	IBM	Thomson CSF
BAE Systems	Litton	TRW

Figure 1. *Organizations participating on the CMMI Steering Group and Product Development Team*

A CMMI design goal is to integrate disciplines, starting with existing capability models and eliminating inconsistencies and duplication to streamline and reduce the cost of model-based process improvement, and increase the return on investment.

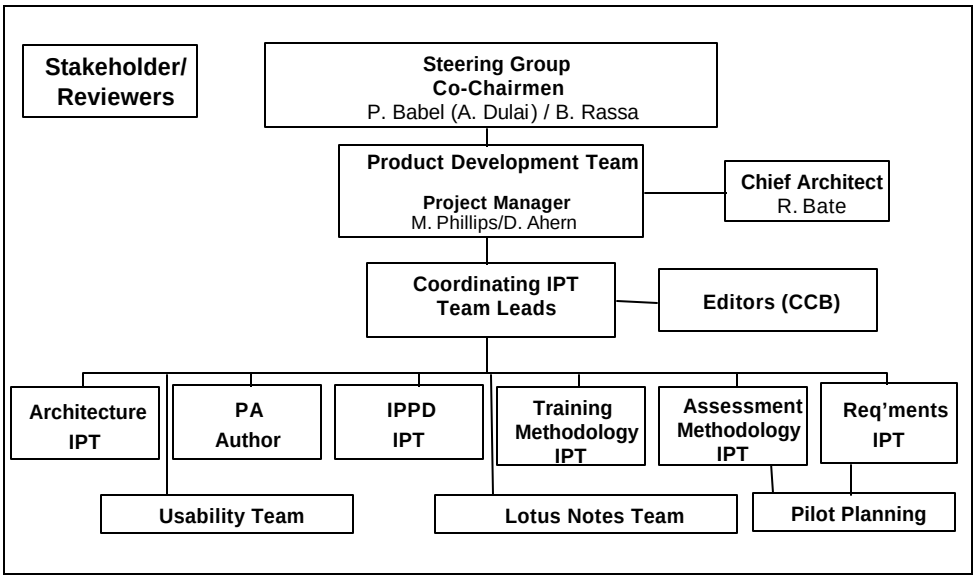
Sponsors

The Department of Defense's Office of the Secretary of Defense for Acquisition and Technology, in conjunction with the National Defense Industrial Association Systems Engineering Committee, initiated the CMMI Project in January 1998 as a collaborative effort among government, industry and the SEI. Organizations participating in the CMMI Project are listed in Figure 1.

CMMI Project Organization

The project organization (See Figure 2.) consists of a steering group and a Product Development Team (PDT) led by the project managers and organized into Integrated Product Teams, with a Chief Architect to ensure the product line's architectural integrity. The steering group provides direction to the project, and developed the requirements specification. Additional duties include configuration control, status tracking and reporting, product approval, issue resolution, transition planning and support, and information dissemination. The PDT is responsible for managing and developing the CMMI products, as well as pilot testing. There is a Stakeholder/Reviewer Group

Figure 2. *CMMI Project Organization*



focused on providing review and comment on draft products and recommending improvements. This group also nominated candidates for pilot testing. PDT and Group members are drawn from government, industry, and the SEI.

Scope

The initial CMMI product suite, as defined by the CMMI requirements specification, focuses on integrating systems engineering, software engineering, and Integrated Product and Process Development (IPPD) best practices to produce equivalent process maturity models in staged and continuous representations.[1] The sources for the CMMI best practices are the SW-CMM v2C, EIA/IS-731 SECM, and the IPPD CMM draft v0.98, from which best practices are being culled and coalesced into common core- and discipline-unique practices and process areas. In addition, integrated training and an assessment methodology are being developed using legacy assessment materials from the source models. In the future, additional related disciplines, such as security engineering, can be added to the CMMI product suite in accordance with a process defined by the CMMI Steering Group. The CMMI scope and concept are illustrated in Figure 3.

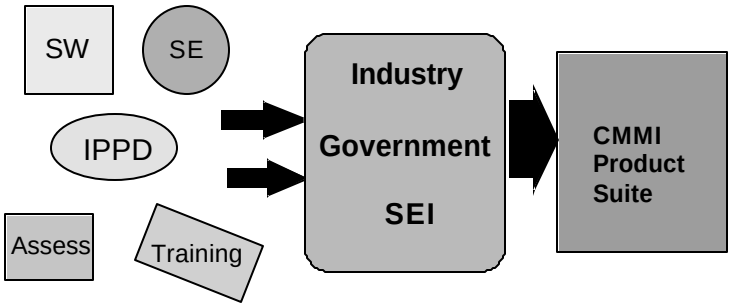


Figure 3. *CMMI Scope and Concept*

What is unique about CMMI?

The CMMI product suite is being structured to integrate multiple disciplines, providing consistent process improvement guidance across related disciplines. The CMMI framework is the backbone of the product suite architecture that implements a product-line approach to enable production of the products from a core asset base. This should result in substantial economies for users, in contrast to working process improvement one discipline at a time in independent efforts, as done in the past for each of the existing single-discipline models. In addition, using the product-line approach should yield significant savings as new disciplines are added over time, and organizations need only adopt a *delta* (for a new discipline) to a CMMI product already in use.

Legacy models as well as other process improvement assets (e.g., training and assessment methods) are being leveraged in the construction of the CMMI product line for generating process improvement tools that can be evolved. In accordance with product line practices, the approach taken is to form new CMMI products using common assets for the disciplines being integrated. This is accomplished by taking applicable components from the asset base, tailoring them as necessary, and assembling them under the umbrella of the CMMI framework, the common, product-wide architecture. Components of the Framework are depicted in Figure 4.

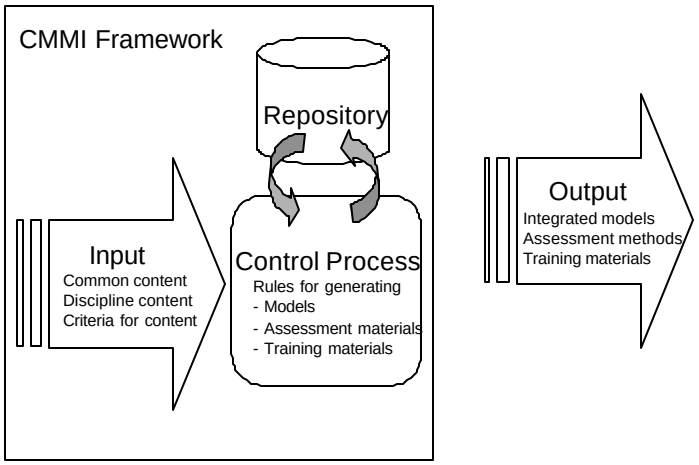


Figure 4. *CMMI Framework Components*

Model Activities/	Process/ Focus Areas	Goals/ Themes	Practices
SW-CMM v1.1	18	52	150
SW-CMM v2 Draft C	19	62	318
EIA/IS-731	19	77	383
IPD-CMM v0.98	23	60	865
CMMI-SE/SW v0.2	24	80	431

Figure 5. *Initial Size Comparison of CMMI-SE/SW Model to Legacy Models*

Figure 5 shows a quantitative view of the evolving CMMI model for systems and software engineering, in contrast with the source models. This figure illustrates the comparative size of the models in terms of process or focus areas, goals or themes, and practices or activities. Noteworthy is the significant consolidation and integration of process areas, goals and practices in the CMMI model in comparison with the source models.

CMMI Product Suite

The CMMI product suite will consist of:

- Integrated models for systems engineering, software engineering, and IPPD. Systems and software engineering are addressed in one model; IPPD concepts will expand this model to integrate IPPD processes, goals and accepted best practices. Each model will be produced in two representations to facilitate transition to CMMI: staged and continuous. However, equivalence between the two representations is being defined to achieve parity during assessments so that the results will be consistent, independent of the representation used.
- Assessment method and instruments for the models. A comprehensive assessment method, the Standard CMMI Assessment Method for Process Improvement (SCAMPI), is being developed to meet the Assessment Requirements for CMMI (ARC). Classes of assessment methods, satisfying subsets of the ARC, are being defined. Assessment data collection methods and tools (e.g., questionnaires) and recommended assessment team qualifications are being developed in conjunction with the assessment method description.
- Training products supporting the models and assessment method. Training will include CMMI model and assessment method training for assessment teams and lead assessor training. Like the CMMI model and assessment method, CMMI training is being developed using a product-line approach.

- Common glossary. Starting with the CMMI requirements specification, common terminology will be defined and used throughout CMMI products.
- Tailoring criteria for CMMI products. These address tailoring of the model and the assessment method. Model tailoring is the use of a subset of the model for purposes of making it suitable for a specific application; assessment method-tailoring addresses the selection of assessment options for use in a specific instance. Model tailoring criteria includes using the model for process improvement and benchmarking.
- A framework for generating CMMI products. This framework is designed to provide an internally consistent set of common elements applicable to any discipline that will be included in any CMMI product. The process for producing CMMI products from the framework is illustrated in Figure 6. The framework is designed to facilitate the addition of related disciplines to the CMMI product suite.

Anticipated Benefits

The greatest benefit for CMMI product-use should be improvement in business performance, as has been demonstrated via use of other maturity models like the SW-CMM. Though unique to an organization's business objectives, performance improvements may include improved profitability, improved win rate for new business and improvements to productivity, quality, and cycle time. In addition, the CMMI products are expected to provide a number of other benefits to users:

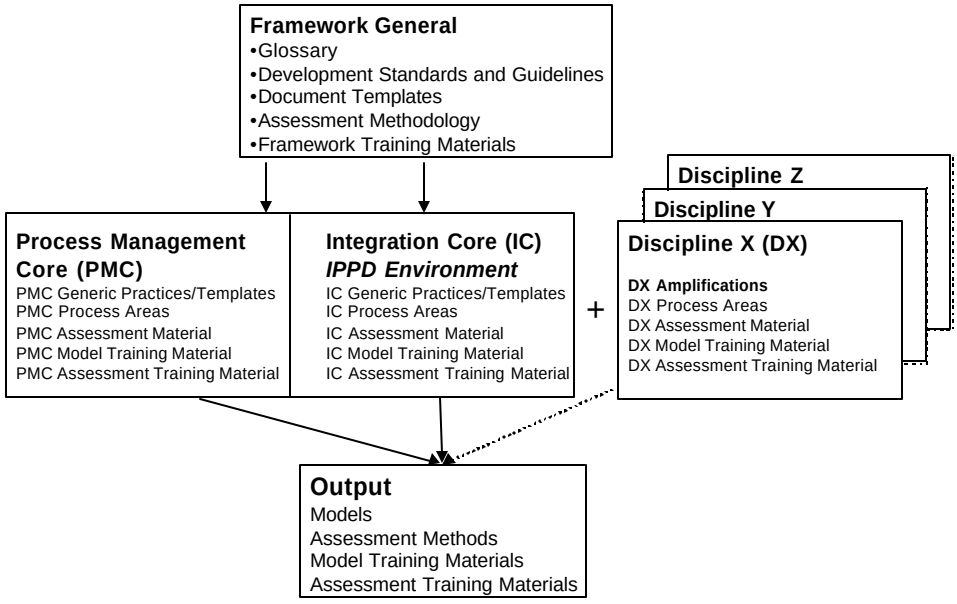
- More efficient, consistent and effective process assessment and improvement across multiple process disciplines.
- More effectively integrated processes, initially systems engineering, software engineering and IPPD.
- Reduced training and assessment costs.
- A common, integrated vision of process improvement across an organization
- An evolutionary process improvement growth path, allowing for incremental addition of new disciplines to the CMMI product line.

The CMMI product suite will allow a long-term process improvement strategy to be formulated, using a single, consistent CMMI product suite. This will facilitate seamless, incremental adoption of additional disciplines over time.

Release Plans

Public review of the CMMI-Systems/Software Engineering (CMMI-SE/SW) v0.2 concluded in November 1999 with nearly 3000 comments received. Pilot assessments began in November. The CMMI-SE/SW v1.0 and CMMI-SE/SW/IPPD v1.0 models are scheduled for public release in June-August 2000. An additional release, CMMI v1.1, is planned for August 2001 to provide additional refinement and update based on the continuing CMMI pilot program.

Figure 6. *Process for producing CMMI Products from the CMMI Framework*



What is different for SW-CMM, EIA/IS-731 users?

CMMI provides a single, integrated model for systems and software engineering process improvement. Users of either the SW-CMM v1.1 or EIA/IS-731 find:

- Additional process areas.
- Additional practices.
- Staged and continuous representations.
- Capability level goals, mapped to institutionalization practices, in the staged representation.

An example of a new process area for SW-CMM v1.1 users who are not deploying EIA/IS-731 is the product verification process area that is included in the CMMI-SE/SW v0.2 model. Though *verify system* is a separate focus area in EIA/IS-731, verification is not a process area in SW-CMM v 1.1. Similarly, measurement and analysis is a CMMI-SE/SW v0.2 process area, but is not included as a separate focus/process area in either SW-CMM v 1.1 or EIA/IS-731.

Users of the CMMI-SE/SW model will note one capability level goal in each process area. Initially introduced in SW-CMM v2 Draft C, the goal addresses whether the in-use process achieves the institutionalization activities related to the applicable maturity level. The intent is to capture institutionalization explicitly in rating during an assessment. Although these goals are new, they should have minimal effect on organizations now rigorously applying institutionalization practices.

The assessment method for CMMI is defined based on the CMM-Based Appraisal for Internal Process Improvement and EIA/IS-731.2, the SECM Appraisal method.

What is the best CMMI transition plan?

The transition plan for each organization will be unique, depending on its process maturity, phase in the process improvement life cycle, and business objectives. For example, a mature organization that recently completed assessments using SW-CMM v 1.1 and EIA/IS -731 may choose to define its updated process improvement plan using the CMMI-SE/SW model. In contrast, an organization with a less mature process, (e.g., a SW-CMM 1.1 Level 2 objective

by the end of 2000) may decide to complete its process improvement and formal appraisal plans before establishing a CMMI transition plan. All transition plans should address the mitigation from legacy models to CMMI within a three-year period after CMMI v1.0 is released, since this is the time frame when legacy models are slated for sunset.

Formulating a transition strategy that addresses the approach for adopting a CMMI model is critical to a successful transition to CMMI. Such a strategy would include a definition of how a CMMI model dovetails with the organization's business objectives and process improvement needs. For users of one or more legacy models, preservation of the investment to date in process improvement will be a key ingredient in the strategy, as will establishing the right level of sponsorship for the CMMI process improvement effort. Organizations using one legacy model will likely need to identify a new, higher-level executive to sponsor a broader (e.g. SE/SW/IPPD) enterprise-wide process improvement effort.

Other considerations for a CMMI transition strategy are establishing buy-in, creating/extending the organization's process improvement infrastructure, involving and communicating with customers, identifying training needs, and estimating the budget required for transition and the expected return on investment. Another consideration would be customer expectations for process maturity and/or improvement.

Formulating a CMMI Transition Approach

The first step is to determine the need and support for process improvement. Given the need, organizational sponsorship for the transition, with allocated resources, must be obtained. Following this, an assessment could be performed to identify the scope and nature of the changes required to adopt a CMMI model. Organizations using legacy models could conduct an informal assessment against a CMMI model, potentially with outside support, or may decide to perform a simple analysis of the changes required. The CMMI Project is producing mappings from legacy models to/from the CMMI models as an aid in doing this. These mappings can be used for an initial gap analysis.

The next step is to establish action plans for implementation, including training, tools, and infrastructure focused on the changes involved from legacy models. Deployment would typically start with the organization's set of standard processes, followed by project processes. After pilot use, specific projects could be identified to apply the improved process and measure its effectiveness and contribution to product and process improvement, with focus on the business parameters identified for improvement by the organizational sponsor. Each use of the CMMI products should be followed by a lessons learned phase during which experiences are compiled and communicated to improve subsequent applications.

Risks in Transitioning to CMMI Products

As a new product suite, the CMMI products have had limited use, primarily on pilots, and data are sparse. However, the CMMI models are well-grounded in proven, publicly accepted practices. A plan for systemically collecting quantified improvements in business performance, including quality, productivity, cycle time and customer satisfaction, and secondary benefits such as improved morale, reduced attrition, and decreased overtime, is needed to support the business case for transition to CMMI products. An additional risk is the potential unavailability of CMMI transition products needed by an organization. Since products like training are being developed incrementally, concurrently with the models, they will initially be available only in pilot form, followed by full rollout by the CMMI product suite steward, the SEI, and its transition partners.

How can I obtain more information on CMMI?

Information on the CMMI project and products can be obtained at www.sei.cmu.edu/cmm/cmms/cmms.integration.html

Note

1. See pages 5 and 6 for more on this.

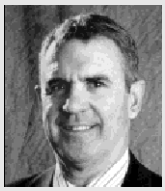
About the Authors

Joan Weszka is manager of Process and Program Performance in the Systems & Software Resource Center at Lockheed Martin Mission Systems. She has more than 25 years of experience in software and systems engineering, and program management. At IBM, she held management and technical positions in systems development of commercial and government large-scale, real-time systems. Weszka is a member of the CMMI Steering Group, and previously chaired the Enterprise Process Improvement Collaboration Steering Group and the SEI's SW-CMM Advisory Board.

700 North Frederick Ave.
Gaithersburg, Md. 20879-3328
Voice: 301-240-7013
Fax: 301-240-7009
E-mail: joan.weszka@lmco.com

Phil Babel is a senior associate with Dayton Aerospace Incorporated. He retired from the U.S. Air Force, in 1999, after 34 years with the Aeronautical System Center, where he was a technical advisor for embedded computer systems software. From 1998 until retirement he served as chairman of the CMMI Steering Group. He has a bachelor's degree in electrical engineering from the University of Detroit and a master's degree in computer and information science from Ohio State University.

4141 Colonel Glenn Highway, Suite 252
Dayton, Ohio 45431
Voice: 937-426-4300
Fax: 937-426-1352
E-mail: Phil.babel@daytonaero.com



Jack Ferguson is the Director for Software Intensive Systems in the Office of the Deputy Under Secretary of Defense for Science and Technology. Prior to this , he was CMMI Project Manager at SEI. He has spent more than 30 years in technical, managerial, and teaching capacities, primarily with the DoD space program and software management. He received the U.S. Air Force R&D Award for his work on GPS spacecraft, and was head of the SEI Joint Program Office. He has a doctorate degree in aerospace engineering from the University of Texas.

1931 Jefferson Davis Highway, Suite 104
Arlington Va. 22202
Voice: 703-602-0851
Fax: 703-602-3560
E-mail: FERGUSJ@acq.osd.mil

A Configuration Manager's Perspective

by Ronald Starbuck
Output Technology Solutions

The Software Engineering Institute (SEI) dynamically broadened the Software Capability Maturity Model (SW-CMM) into an integrated suite (CMMI) of process models that now span the enterprise. This expanded comprehensive approach to software engineering through processes was the result of merging the updated unreleased SW-CMM with the other best features of software and system-related disciplines to form a single product maturity framework. This article looks at the CMMI from the perspective of a Configuration Manager to see what its influences were to Configuration Management (CM) since the release of the SW-CMM Version 1.1.

Background

The integration of the CMMI resulted in combining Version 2.0 of the SW-CMM, Draft C with the Electronic Industries Alliance (EIA) Interim Systems Engineering Capability Model Standard (SECM) and the CMM Integrated Product Development, Draft Version 0.98 (IPD-CMM) [1]. (See Figure 1.) The result is a suite of products in which CM continues to be a major building block in software/systems development. It continues to ensure the product life cycle commonality, compatibility, and consistency. Version 2, draft C of the SW-CMM was the major contributor to CM in the CMMI. It evolved from what has become the de facto standard for assessing and improving software engineering processes the SW-CMM, Version 1.1. This updated version was never released but included more than 180 user requests from lessons learned in SW-CMM implementation, defining a better understanding of higher software maturity, and achieving better consistency with the other software industry standards and terminology. The staged presentation of the CMMI is used for comparison in this article as its framework best reflects the architecture typified by the SW-CMM [2]. An organizational maturity level is established by a maturity framework structure in which the goals of a set of process area are attained.

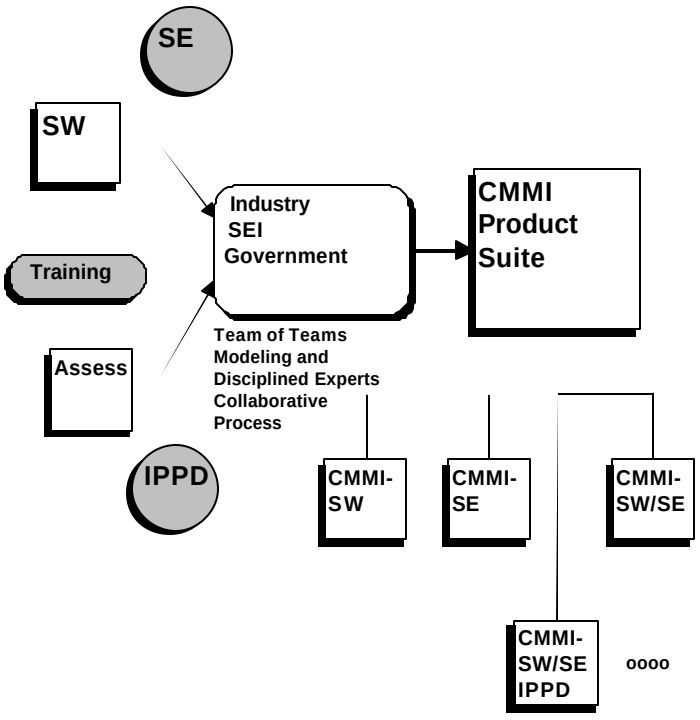


Figure 1. *Model's Creation*

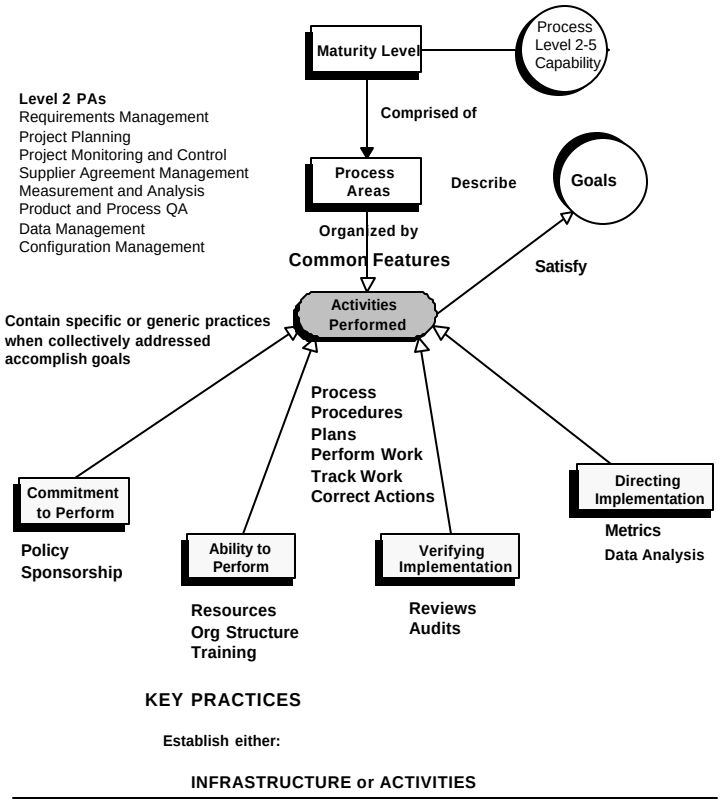


Figure 2. *Architecture of the Model*

Overview

The CMMI continues to embrace the same time-proven structure that evolved the SW-CMM, Version 1.1 into a world class model for software development. The staged presentation's structural design is still composed of five ascending levels of process maturity supported by defined Process Areas (PA) that collectively achieve a maturity level (see Figure 2). These clusters of related practices are performed collectively to achieve a set of objectives for each maturity level. They continue to be called common features, predefined attributes that signify whether the implementation and institutionalization of a process area are effective, repeatable, and lasting. A top-level check of the staged presentation of the model reveals the obvious changes since Version 1.1 of the SW-CMM:

- Most notable is the Maturity Level 2 name changed from "Repeatable" to "Managed."
- Removal of "Key" from "Key Process Area (KPA)."
- A different set of PAs comprises the CMMI. Two have been added to the six in the SW-CMM for Level 2. They are Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management,

Measurement and Analysis, Product and Process QA, Data Management, and Configuration Management.
- The five Common Features now include a new key practice, Directing Implementation, that replaces the SW-CMM key practice of Measurement and Analysis. The five common features implementing the CMMI are now commitment to perform, ability to perform, activities performed, directing implementation, and verifying implementation.[1]

Detailed Look

It is obvious these aggregate changes in the integrated model will have an impact on the way in which CM is to be done. Today's CM encompasses a broad spectrum of interrelationships and associations with all the disciplines and business dependencies that make up the organization that produces the software and will require some retooling. A closer look at what these changes are to the Configuration Management Process Area follows:

The CMMI Maturity Level's "Repeatable" and "Key Process Area" names were changed, respectively, to "Managed" and Process Area." They should not have any effect on CM; however, they definitely entail a cultural transitional learning curve as the SW-CMM names have evolved to signify world-class definitions people have come to know as the Capability Maturity Model. In general some Process Area's are different in the CMMI as it modified some of the existing SW-CMM Key Process Area's and added two additional Process Areas, to bring the total to eight defining CMM Level 2 in the SW-CMM. It also changed the purpose statement of Configuration Management, "to establish and maintain the integrity of the products of the software project throughout the projects life cycle," [2] to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits [1]. They are:

- Combining and modifying the SW-CMM KPA, "Software Quality Assurance" (SQA) into "Product and Process Quality Assurance" (PA) seems to relax the former KPA of SQA. The purpose of Process and Product Quality Assurance is to objectively review activities and work products for their adherence to applicable require ments, process descriptions, standards, and procedures, and communicate the results to staff and management. It will have a slightly different role of visibility through objective reviews of the way products are developed.
- The "Data Management" PA is new. It provides administrative management of appropriate project data, both deliverable and nondeliverable project data and maintain its availability to the project staff and stakeholders. This and the CM PA are closely related. It addresses other data and focuses on data needs, the data development schedule requirements, date acquisition and control, and access to data. CM is focused on the rigorous control required for the technical work products.
- Changing the name from SW-CMM KPA "Software Project Tracking and Oversight" to "Project Monitoring and Control" better describes the PA's purpose—to provide adequate visibility into the progress of the project so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- Changing the name from SW-CMM KPA "Software Project Subcontract Management" to Supplier Agreement Management" is just a name change. The purpose of Supplier Agreement Management is to manage the acquisition of products and services from sources external to the project to provide adequate visibility into a project's progress. This is so appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- The "Measurement and Analysis" PA is new. Its purpose is to develop and sustain a measurement capability in support of management information. This PA was derived from the Measurement and Analysis Common feature to a PA as a definite lesson learned and as a requirement of the SECM. It centralizes organizations to implement measurement easier than if the equivalent practices spread across multiple PAs, as was done in the SW-CMM. It also is due in part for Level 4 requirements, [3]. Its addition to the model will add to the importance of configuration status accounting data and practices in some fashion.
- Project Planning's purpose is to establish and maintain plans that define project activities. While the KPA of SCM in the SW-CMM specifically identified a Software Configuration Management Plan, the CMMI relaxed this to cover the practices for performing CM functions. However, it better standardized what content is needed for establishing and maintaining plans to control the project.
- The CMMI uses "Directing Implementation," a new key practice based on the SW-CMM "Measurement and Analysis." Significantly, this key practice now implements management and analysis rather than saying they need to be done in the SW-CMM. This change enhances the key practice with action on what to do.

Polished Purpose Statement

The CM's purpose has evolved from SW-CMM Version 1.1, which was "to establish and maintain the integrity of the products of the software project throughout the project's software life cycle." Its purpose has been redefined as establishing and maintaining the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits in the CMMI. This definition to a configuration manager is classic configuration management at its best. It establishes and maintains the integrity of work products as they evolve through the full life cycle to ensure the bottom-line capability of CM to remanufacture products, the timeless and sometimes forgotten reason for doing CM. To me, CMMI configuration management is more consistent with its ancestry rooted in such government software standards as MIL-STD-483A, "Configuration Management Practices," released in 1970. It is a time-proven descendant of today's modern configuration management, but now process-oriented in the CMMI—today's trend for the application of CM.

Concerns

The latitude given by the CMMI to tailor how you do business can allow some flawed interpretations by inexperienced configuration managers. In relaxing language, these configuration managers could take the acceptable alternatives road allowed by two of its key practices.

The first concern is the ability to perform (AB1.) [1], "Plan the process, establish and maintain the requirements, objectives, and plan for performing the CM process." This means a configuration management plan (CMP). The alternative is a less acceptable set of processes. Without a CMP roadmap, these processes are not choreographed as to the order of steps from the development process until the final release of the product. A CMP purpose is to put all of these process sets into their proper life-cycle perspective [4].

The second concern is in the activity performed (AC5.) [1], "Establish and maintain the Change Request System." Your change management system should include a documented Configuration Control Board that fits your organization's needs—which the CMP should define. It provides a disciplined perspective of what is to be changed from a board of subject matter experts that has a big-picture view, or properly addresses what is proposed for change. The full board is used only when necessary, as not all changes have to go through all of the subject matter experts for a decision. The alternative to doing anything less is a subordinate way of making changes [4].

CM Conclusions

As re-instrumented in the CMMI-SE/SW, the staged representation continues the CM's time-proven functionality to fashion and maintain the necessary integrity to develop and reproduce software work products from baselines. The CMMI is more process-ordered than its predecessors and leads the trend for today's implementation of CM. It is one that features process dependencies or relationships with all of the disciplines involved in software development. Implementation success, as always, will depend on the sophistication of CM that the organization has already reached, and the experience level of its Configuration Manager. Less mature organizations could misinterpret some CMMI-acceptable alternative practices, resulting in something less than the best way to do things.

References

1. Public-release draft, *Staged Representation for CMMI*-SE/SW, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. 1993.
2. Paulk, M. et.al, *Capability Maturity Model for Software,* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. 1993.
3. Lucero, Scott, *Insight, The Army's Software Metrics Newsletter*, Spring 1999, The Capability Maturity Model Integration, An Interview with Bruce Allgood and Lt. Col. Jarzombek. Vol. 3, No. 4. [Ed note: reprinted this issue, page 4.]
4. Starbuck, Ronald, From the From Line, *Software Testing & Quality Engineering,* Vol. 1, Issue 6, 1999.

Note

1. See page 6 for more on this.

About the Author

Ronald Starbuck is a configuration manager at Output Technology Solutions, where he is involved in improving customer systems' department configuration management and software development process infrastructure. He has spent 21 years in government positions and agencies, such as lead programmer for the Navy's P-3 Orion Weapons Simulator., configuration manager at the Sacramento Army Depot for test program sets, and a software quality assurance representative for the Defense Logistics Agency

Output Technology Solutions
Attn: Ron Starbuck, Mail Stop 4150
1102 Investment Blvd.
El Dorado Hills, Calif. 95762
E-mail: Ronald_Starbuck@billing.com
Voice: 916-941-4250
Fax: 916-941-4064

Combined Automated Lessons Learned (CALL) Information Center

Lessons Learned are positive and negative experiences of value that are documented by a researcher, validated by a subject matter expert, and uploaded into an infobase maintained by NAWCAD 4.1.6.2 at PAXRIVMD. Lessons Learned are intended to inform people and influence intelligent decision making at all levels of business and all phases of production. The infobase is accessible on a Web site and access is authorized for military, government civilian, government contractor, and select academies of higher learning. The infobase contains about 5,000 lessons that cover 43 impact areas such as Acquisition Reform, Best Practices, Contract Administration, Design Engineering, Human Factors, Maintenance Engineering, Reliability, Safety, Test and Evaluation and Training on all Aerospace Programs/Platforms and Support Equipment. Lessons written for the Navy, Air Force and the FAA are present in the infobase. The World Wide Web site is located at www.nawcad.navy.mil/call

Points of Contact

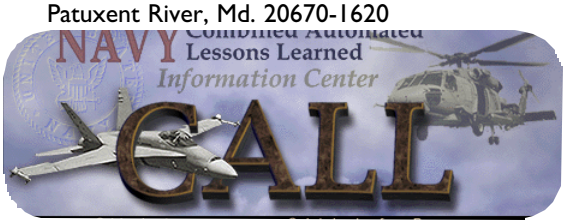
Dawn Hassinger, CALL Manager
Voice: 301-342-2170
Fax: 301-342-4782

Ron French, CALL Administrator
Voice: 301-342-4284
Fax: 301-342-1247

Sal Pistachio, CALL Researcher
Voice: 301-342-4286
Fax: 301-342-1247

Mailing Address

Ron French
Naval Air Warfare Center Aircraft Division, Bldg. 2102
22689 Saufley Road, Unit 1
Patuxent River, Md. 20670-1620



Transitioning from EIA/IS-731 to CMMI

By Aaron Clouse
Raytheon Co.

Curt Wells
i-metrics

The Capability Maturity Model® Integration (CMMI) project has integrated the Capability Maturity Model for Software (SW-CMM®) Version 2C, Electronics Industry Alliance (EIA/IS-731) Systems Engineering Capability Model Version 1.0, and the Integrated Product Development Capability Maturity Model (IPD-CMM) Version 0.98a. For the systems engineering community that has been using EIA/IS-731, this integration of CMMs has resulted in many changes to the practices and amount of information contained in the model. This article addresses the changes for each EIA/IS-731 Focus Area.

The CMMI project integrated the SW-CMM® Version 2C, [1] EIA/IS-731 Systems Engineering Capability Model Version 1.0 [2], and the IPD-CMM Version 0.98a [3] into a single framework that can be used by any organization whose processes involve developing a product or service for process development and improvement. In developing the CMMI, authors are constrained by size and complexity considerations relative to the activities/practices that can be included from source models. Mapping files can be used to help the user community evaluate whether a reasonable suite of practices was included from the source models, and to aid in transition from the source models to CMMI. The differences between CMMI Version 0.2 and EIA/IS-731 are described in this article. The impacts described may change when EIA/IS-731 is compared to CMMI Version 1.0.

Architecture Comparison

EIA/IS-731 is organized by Categories, Focus Areas, Themes, and Specific Practices. These, along with Generic Practices and Generic Attributes, provide the model used for assessments (See Figure 1). The only informative material in EIA/IS-731 is in the Descriptions and Notes found in Generic Practices, Generic Attributes, Focus Areas, and Themes. Specific Practices do not provide any informative material.

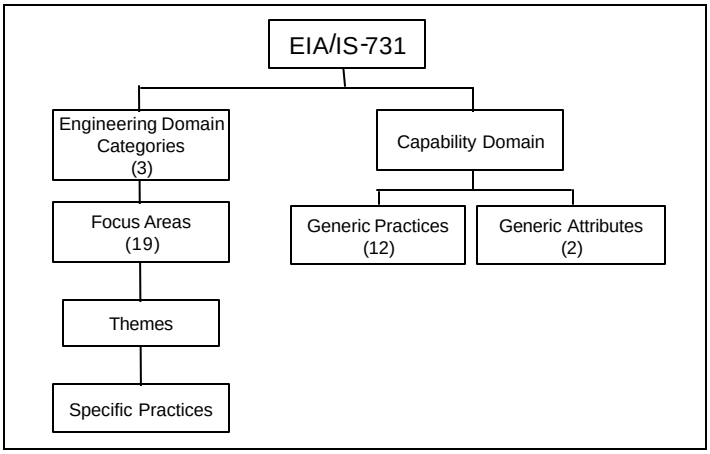


Figure 1. *EIA/IS-731 Architecture*

The CMMI is similar to EIA/IS-731. The primary difference between the models is the amount of informative material included. Compare Figure 2, CMMI Architecture, with Figure 1, EIA/IS-731 Architecture. CMMI has fewer normative and expected elements than EIA/IS-731. EIA/IS-731 has 19 Focus Areas, 77 Themes and 381 Specific Practices. CMMI has 28 Process Areas, 92 goals and 221 Specific Practices. CMMI has 22 Generic Practices while EIA/IS-731 has 12.

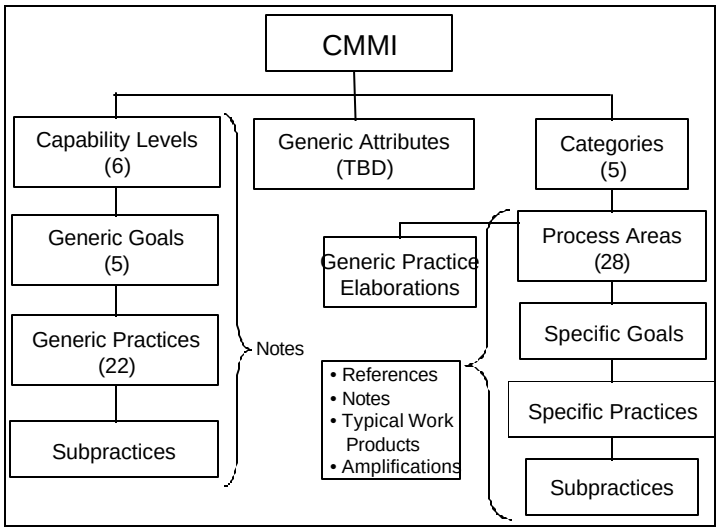


Figure 2. *CMMI Architecture*

Capability Levels, and Categories of Process Areas organize CMMI. Generic Attributes are under study and may be added to the model. The model's Specific Practices are organized by Categories, Process Areas and Specific Goals. Generic Practices are organized by Capability Levels and Generic Goals.

The CMMI provides informative material in several formats.

- References provide links to other process areas. There are two kinds of references in CMMI, Refer to and Use. "Refer to" means go to the referenced Process Area (PA) for more information on the topic. "Use" means use the practices in the referenced PA to perform the practice. Both references are informative only, no required or expected inference is intended.
- Notes are allowed for every level of the architecture to explain the intent and provide examples of the model component.
- Specific and Generic Practices may have Subpractices to provide more detailed level of informative material for the practices.
- Each Specific Practice and Subpractice may have amplifications to provide information specific to a discipline.
- CMMI contains example Work Products to help interpret the practice.
- Generic Practice Elaborations provide informative material. They are organized by PAs because each Generic Practice is used for all Process Areas.

The CMMI documentation provides an introduction, model structure description, and a discussion of how to understand the model, how to use it, references, acronyms, and a glossary. These features provide additional informative material for the user. See Figure 3, CMMI Document Structure.

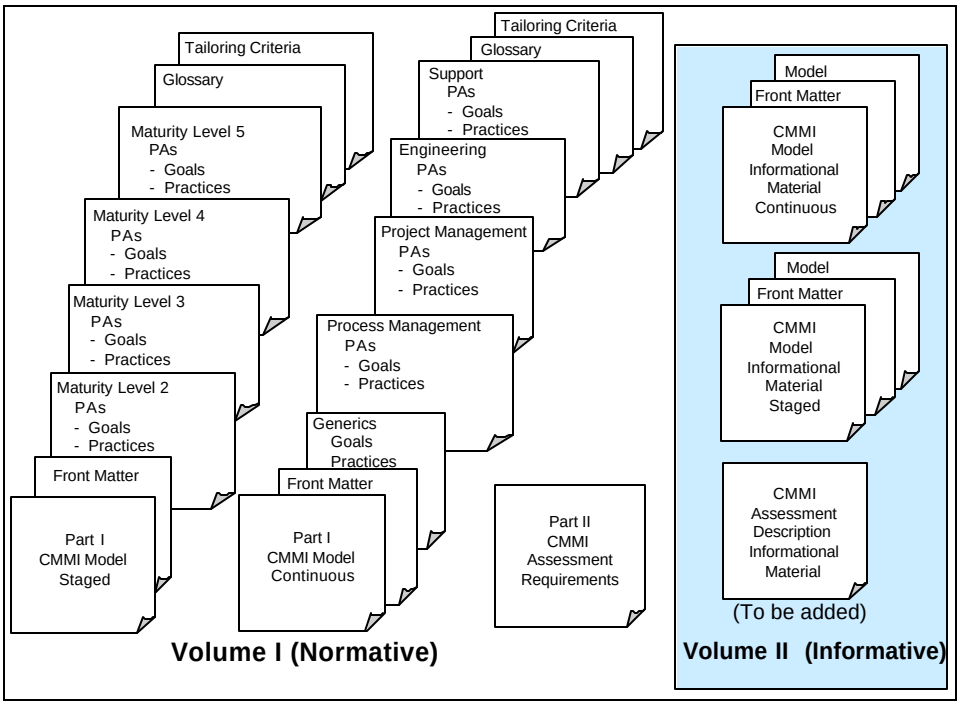


Figure 3. *CMMI Document Structure*

Comparison of Models

The following sections describe how EIA/IS-731 ('731) maps to and from CMMI Version 0.2. Please note that several changes are under investigation that may result in changes to the CMMI Version 1.0. The detailed mapping will be updated after its release. The detailed mapping can be accessed at www.sei.cmu.edu/cmm/cmmi/comm/map731.html

Technical Category Focus Areas

In developing the CMMI, there were a number of opposing forces that had to be overcome in integrating the systems and software models. '731 has six Focus Areas (process areas) that address engineering, while the SW-CMM has one Process Area (Software Product Engineering) to cover the same material. '731 has a large number of practices per process area, while the SW-CMM is relatively sparse in the number of practices (activities) per process area. The integration compromise resulted in five process areas for engineering, similar to '731. However, the number of practices was reduced by about half, relative to '731. As CMMI is released for comment and public use, several questions, important to the ultimate success of CMMI, will need answers from the systems and software community. Will the software community find the increase in the number of engineering process areas and practices helpful or burdensome? Will the systems engineering community find the elimination of the Define Technical Problem Focus Area and the reduction in the number of practices acceptable? Are '731 practices in CMMI the appropriate ones?

The representation of each of the '731 Technical Category Focus Areas in CMMI, and some effects of transitioning, are briefly discussed below. Recommendations, related to model transition, represent the authors' judgement.

Define Stakeholder and System Level Requirements CMMI Customer and Product Requirements

The CMMI Customer and Product Requirements (CPR) PA includes the two themes from EIA/IS-731, Define Stakeholder and System Level Requirements, under Goal 1. Goal 2 of the CMMI CPR PA includes selected practices from Problem Refinement Theme of the '731 Define Technical Problem Focus Area. (The Define Technical Problem Focus Area does not exist as a PA in CMMI. Several of its practices are distributed to other CMMI PAs.)

The CMMI CPR PA addresses, in less detail, the following '731 concepts:

- Collecting stakeholder needs and the related advanced concept of eliciting stakeholder needs.
- Converting needs to requirements.
- Obtaining agreements on the requirements with customers.
- Validating requirements.
- Refinement of requirements.

The CMMI includes the '731 base-advanced practice pair that addresses collecting stakeholder needs and the advanced practice of stakeholder needs elicitation, although some clarity in the distinction between the concepts is lost. Four '731 practices dealing with the analysis, prioritization, and reporting on stakeholder needs are not included in CMMI. The CMMI combines three '731 practices that deal conceptually with validating requirements into two practices, with some improvement in consistent use of the term *validate*. CMMI does not include the practice on identifying key requirements. Process owners may want to note this one. Identification of key requirements (drivers) is often valuable in controlling system development cost. A number of the '731 practices left out of CMMI may be needed in organization process descriptions, depending on the organization's business needs. In some cases, the CMMI version improves on the clarity of an EIA/IS-731 concept or practice.

Define Technical Problem Focus Area

Some Define Technical Problem (DTP) Focus Area practices were included in the CMMI Customer and Product Requirements PA; some were included in the Technical Solution PA. DTP addresses logical or functional problem analysis (solution independent) and requirements document maintenance. Fifteen DTP practices were not included in the CMMI. Excluded practices that deserve a close look (for transition considerations) include practices that address identifying key stakeholder requirements, capturing relationships between requirements, and capturing rationale for requirements traceability, derivations, and allocations. '731 practices that deal with reviewing requirements against quality attributes and maintaining the status of requirements are represented in the CMMI Requirements Management PA as (informative) subpractices. The entire '731 Feedback and Verification Theme, relating to involvement of stakeholders in requirements development, is not included in CMMI. Several '731 practices that are not specific practices in CMMI are covered by the CMMI generic practices (Common Features).

Define Solution CMMI Technical Solution

The CMMI Technical Solution process area includes most of the '731 Define Solution practices. It also includes the functional architecture practices from Define Technical Problem and the product construction and supporting documentation activities from the SW-CMM Software Product Engineering process area. A significant '731 practice, not included in the CMMI, is one that addresses review of derived and allocated requirements in the context of operational concept threads and scenarios. As this practice is valuable for evaluating the adequacy of requirements, organizations should consider retaining it as a key process element. Three of the CMMI practices include several of the '731 concepts in bulleted lists. The CMMI practice on development of design alternatives and selection criteria invokes considerations of life cycle cost, performance, complexity, robustness, expansion and growth, cost drivers, technology limitations, sensitivity to construction methods and materials, risk, and evolution of requirement drivers and technology. The CMMI activity on maintaining complete design descriptions calls for including functionality, requirement allocations to design components, operational concepts and scenarios, architectural features, and design decision rationale. The CMMI activity on developing component specifications calls for specifying each design component in terms of allocation of product performance, design constraints, fit, form, and function to meet requirements and facilitate production and derived requirements that address the cost and performance of other life-cycle phases.

These practices embody a lot of material, and although they may prove awkward in assessments, the material should be valuable for process definition and improvement guidance.

Assess/Select CMMI Decision Analysis/Resolution

All of the '731 Assess and Select practices are included in the CMMI Decision Analysis and Resolution process area. The CMMI Decision Analysis and Resolution process improves on the '731 Assess and Select focus area by adding a practice on establishing and using criteria to determine which issues to subject to a formal decision analysis and resolution process.

Without this, the question nearly always came up in assessments as to whether all decisions, or some select set, were to be subjected to a formal decision process. The new practice solves this problem by requiring organizations to establish criteria for selecting those decisions important enough to need a formal resolution process.

Integrate System CMMI Product Integration

About half of the '731 Integrate System Focus Area practices are included in the CMMI Product Integration Strategy Process Area. Only two relatively high-value practices are not included in CMMI. One of these is the '731 advanced practice, "develop the integration strategy early in the program."

Organizations should continue to make early integration planning part of their standard process. The other is a practice that addresses formal procedures for coordination of multiteam integration efforts. Of the remaining '731 practices not included in CMMI, some are handled by generic practices and other process areas and the remainder are of relatively low value. The CMMI Product Integration process area includes a new practice, "select the optimum integration strategy." It would be reasonable to expect that the word *optimum* will be eliminated in a future release. The CMMI Product Integration process area adds (relative to '731) a practice on acceptance tests performance and a practice on product packaging and delivery. The presence of the acceptance test practice in the integration process area is puzzling and seems to overlap with material in the Product Verification process area.

Verify System CMMI Product Verification

Relatively high-value practices not included in CMMI are the practices that address validation of verification procedures and support facilities, incremental verification, and review and coordination results with stakeholders. Practices that address these topics are recommended as valuable elements of organizational processes for product verification. A number of other '731 System Verification practices are not included in CMMI: these are moderate to low value, mostly due to CMMI generic practices and some redundancy in '731. Readers should be alert to the use of the term *work products* in the CMMI Product Verification process area. The term appears to include the product(s) delivered to the customer, hence, the overlap mentioned above with the acceptance test practice of Product Integration.

Validate System; CMMI Validation

The CMMI Validation process area has improved on the '731 practice of performing validation, by separating out training, maintenance, and support validation as a separate practice. CMMI does not include the '731 practices on early requirements validation. These '731 practices should be considered for any organization's standard process on validation.

Management Category Focus Areas

The application of advanced practices is the primary difference between '731 and CMMI. '731 defines specific practices by capability level. The SW-CMM uses only *base* practices called activities. The CMMI Engineering process areas use advanced practices, but the Management and Support process areas do not. Many of the advanced practices, those that are at Level 2 or higher, now map to CMMI base practices. This, in effect, raises the bar for the continuous assessments with respect to '731.

The CMMI does this by including many of the '731 practices as subpractices. As shown in the introduction, subpractices are informative only. However, informative material should be used for guidance on process improvement programs. CMMI improves on '731 by providing informative material for this purpose while reducing the number of specific practices assessed.

Each '731 Focus Area in the Management Category is discussed in this section. Unless noted, there is no significant impact in CMMI transition, other than lack of advanced practices. There are practices included in CMMI that are new to '731 users. Table 1 shows the number of these practices by process area.

As the table shows, organizations transitioning to CMMI should consider Quantitative Management of Quality and Process (QMQP), Measurement and Analysis, (M&A) Causal Analysis and Resolution (CAR) and the new practices in risk

Management or Support Process Area Practices That	Acronym	Number Of Are
New To '731 Users		
Organizational Process Focus	OPF	0
Organizational Process Definition	OPD	1
Organizational Training	OT	0
Quantitative Management of Quality and Process	QMQP	5
Organizational Process Performance	OPP	0
Causal Analysis and Resolution	CAR	5
Organizational Process Technology Innovation	OPTI	4
Process Innovation Deployment	PID	2
Project Planning	PP	1
Project Monitoring and Control	PMC	3
Suppler Agreement Management	SAM	1
Integrated Project Management	IPM	1
Risk Management	RSKM	4
Configuration Management	CM	3
Data Management	DM	1
Process and Product Quality Assurance	PPQA	0

Table 1. *CMMI Practices New to EIA/IS-731 Users*

management. The new QMQP practices deal with statistically managing the subprocesses. M&A defines the process that should be used to establish a measurement system. This process is not covered in '731. CAR practices deal with determining and addressing root causes of defects.

Plan and Organize

Plan and Organize practices are included primarily in CMMI Project Planning. Some practices are included in Integrated Project Management and two are in Measurement and Analysis. Six '731 practices are not included in CMMI and two are covered by CMMI generic practices. One practice not included addresses designating a systems engineering first-line manager for negotiating technical commitments. This is an important systems engineering role and should be considered when assigning responsibilities. Three practices address the statement of work (SOW) and the work breakdown structure (WBS). The WBS is addressed in CMMI Project Planning in a more general manner. '731 has three levels of capability addressing the SOW and WBS. Another practice not addressed directly is developing top-level schedules for the remaining life cycle phases of the program. CMMI addresses defining the project life cycle in Project Planning. This, along with the planning generic practice and the Establish and Maintain Schedules Practice adequately covers this practice. The last practice not addressed in CMMI addresses clear lines of responsibility and authority between systems engineering and program management. The Assign Responsibility and Planning generic practices and the overall Project Planning PA imply this practice.

Monitor and Control

Monitor and Control practices are included in CMMI Project Monitoring and Control, Integrated Project Management and Measurement and Analysis. Two practices in Theme 2-2.1, Degree of Formality, are not included in CMMI that address the degree of oversight needed, and establishing criteria for program evaluation. Establishing the criteria is important and should be a part of a measurement or monitoring and controlling process. CMMI addresses determining objectives but not specific criteria. One practice is covered by a CMMI generic practice.

Integrate Disciplines

Integrate Disciplines' Practices are included in CMMI Integrated Project Management. Three '731 practices are not included in CMMI. One addresses training that the CMMI Organizational Training PA can address if there is a need for interdisciplinary cooperation. Another practice not included is a Level 4 practice that addresses modeling communication skills and interdepartmental cooperation of upper management. This is an advanced practice and could not be placed in CMMI with the current architecture. Another Level 3 advanced practice addresses establishing a mechanism to ensure compliance with commitments. Both of these should be considered when defining integrated disciplines processes. One practice is covered by a CMMI generic practice. One of the practices is covered by a practice in an IPPD process area, Integrated Team. Only four Integrate Disciplines specific practices (SPs) are included in CMMI as SPs. The remaining '731 practices map to CMMI informational components.

Coordinate with Suppliers

Coordinate with Suppliers practices are included in CMMI Suppler Agreement Management. Six practices are not included in CMMI. The first addresses selecting suppliers based on inputs from the systems engineering team leader. The integrated model addresses this by the Assign Responsibility and Planning generic practices. A Level 4 '731 specific practice builds on this concept by having systems engineering participate in the plans, process, and product standards the suppliers use. Another practice is a Level 4 specific practice in '731 and addresses involving the supplier early in the program to assist in requirements development. This is not addressed in CMMI but should be considered when appropriate. The other practices address providing feedback to the suppliers and having a mechanism for assuring that they follow their processes. These are important practices for both the project and the suppliers and should be included in an organization's processes.

Manage Risk

Manage Risk Practices are included primarily in CMMI risk management. One practice is included in Project Monitoring and Control and four are covered by CMMI generic practices. Eleven practices are not included in CMMI. Theme 2-5.5 contains three practices that should be included in a risk management strategy. Theme 2.5-8 addresses communication and coordination of risk status. These practices are not addressed in CMMI but should be considered as an important part of an organization's process. A Level 3 specific practice not

included in CMMI addresses implementing risk management for key processes within the program. CMMI addresses several other aspects of risk but not processes for design, test, manufacturing, etc. CMMI provides adequate coverage of this practice although it is not explicitly stated. Another specific practice addresses assessing risks qualitatively. CMMI has a practice that assesses the likelihood and consequences for each risk that probably covers qualitative assessment of risks. One specific practice addresses reviewing risk analysis. In general, specific reviews are not listed in CMMI. There is also a Level 4 specific practice that addresses using metrics regarding risks to initiate corrective action. There is a subpractice under the CMMI Implement Mitigation Plans specific practice that collects performance metrics on the risk-handling activities. This only partially covers the '731 specific practice. The other specific practice not included in CMMI is a *document* practice. CMMI does not include *document* at the specific practice level but it is implied. This construct may be improved in future releases.

Manage Data

Manage Data practices are included in CMMI Data Management. Two practices are not included in CMMI. One provides a common data management archival and retrieval capability throughout the organization. CMMI is not specific on providing data management capability throughout the organization. The CMMI practices can be applied at any level in the organization. The other specific practice that addresses archiving data efficiently was not included in CMMI because the term *efficiency* is difficult to assess. CMMI tried to avoid subjective words. CMMI generic practices cover three practices. An improvement provided by CMMI is a practice on establishing data privacy and security.

Manage Configurations

Manage Configurations practices are included in CMMI Configuration Management. One '731 advanced practice (2.7-2-3) relating to evaluation of change impact beyond the immediate program is not included in CMMI. This practice should be considered for inclusion in organizational processes.

Ensure Quality

Ensure Quality practices are included primarily in CMMI Process and Project Quality Assurance. One practice is covered by Quantitative Management of Quality and Process. One is covered by Causal Analysis and Resolution and two are covered by CMMI generic practices.

Environment Category Focus Areas

Each '731 Focus Area in the Environment Category is discussed in this section. Unless otherwise noted, there is not any significant impact in transitioning to CMMI, other than the lack of advanced practices.

Define and Improve Systems Engineering Process

Define and Improve the Systems Engineering Process practices are included primarily in CMMI Organizational Process Definition and Organizational Process Focus. Some practices map to CMMI Qualitative Management of Quality and Process, Integrated Project Management, and Organizational Process Technology Innovation. Two practices map to a CMMI generic practice and two are not included in CMMI. One of these practices clearly defines the inputs and outputs of the systems engineering subprocesses. The word *clearly* is subjective and is not included in CMMI. The practices included in CMMI are defined to a level of detail that improves on the '731 specific practice. The other specific practice performs improvement of systems engineering processes in use on programs in at least an informal manner. CMMI Organizational Process Technology Innovation provides several specific practices that address process improvement. This is probably an improvement provided by CMMI.

Manage Competency

The '731 Manage Competency Focus Area is represented in CMMI by the Organizational Training process area. Many of the '731's 30 practices are included as subpractices of the six CMMI practices. The Learning Environment Theme of '731, which addresses methods for creating a learning environment, is not included in CMMI. Inclusion of this important theme in an organization's process is highly recommended. The '731 practice (3.2-4-4) on evaluating alumni capability is not included in CMMI; organizations should consider this practice at higher capability levels. The '731 practice relating to provision of skill and knowledge from outside sources was moved to the Project Planning process area in CMMI.

Manage Technology

Manage Technology Theme 3.3-1 is not included in CMMI. One practice from Theme 3.3-2 is included in CMMI Suppler Agreement Management. Theme 3.3-3 is included in CMMI Process Innovation Deployment, Organizational Process Technology Innovation, and Integrated Project Management. Systems engineers consider this a weakness in CMMI. Future versions of the CMMI may include these practices.

Manage Systems Engineering Support Environment

Manage Systems Engineering Support Environment practices are included in CMMI Organizational Process Definition and four CMMI generic practices. Four practices are not included in CMMI. Piloting new tools is one specific practice that is not included in CMMI. If tools were considered a part of new processes, then this would be addressed by Organizational Process Technology Innovation (OPTI), where piloting is addressed in the "pilot-selected process improvements" practice. Perform cost-benefit analysis for commercial off-the-shelf vs. in-house developed environments is another specific practice not addressed in CMMI. OPTI addresses cost-benefit analysis but does not specifically address this trade-off. The new processes addressed in OPTI could include off-the-shelf systems engineering environment improvements, however. Another practice not included uses the word *maximize*. This is difficult to assess and will not be used in CMMI. The fourth practice not included addresses planning and tracking maintenance of the support environment. The key words in this practice relate to monitor and control maintenance that is not addressed in CMMI. However, a combination of several practices indirectly addresses the concept.

Generic Practices

The CMMI generic practices are similar to those found in '731. It has 12 generic practices while CMMI has 22. In general, the CMMI is more detailed than '731. The SW-CMM common features are the source for several of the generic practices. The CMMI generic practices are grouped under Generic Goals. There is one generic goal for each capability level. The goals are general in nature: "The process is institutionalized as an xxx process." XXX represents the five levels, (i.e., Performed, Managed, Defined, Quantitatively Measured, and Optimizing).

The CMMI generic practices may be grouped within categories in CMMI Version 1.0. The following categories are in use in the CMMI Staged Representation and may be used in the Version 1.0 Continuous Representation:

- Commitment to perform.
- Ability to perform.
- Directing implementation.
- Verifying implementation.

There are several differences between the staged and continuous representations of CMMI GPs. They will not be discussed here.[1] These differences and the grouping of generic practices does not impact the transition from '731 to CMMI. The level of detail in the generic practices statements does impact the assessment effort. This may not be a negative impact because the level of detail does provide more information to the assessors and the organization using the model for internal process improvement activities. '731 users will find the generic practices in CMMI a little easier to use as they are not as compounded (i.e., do not have as many concepts in a single practice) as '731.

Summary

Due to significant differences between EIA/IS-731 and CMMI, a well-planned transition is recommended. Organizations are encouraged to avail themselves of appropriate training and reference resources in planning and effecting the transition. The Software Engineering Institute, the International Council on System Engineering, and the Government Electronics and Information Technology Association are expected to provide transition resources. Organizations should also carefully evaluate the business value of existing practices in their standard processes before making model evolution/transition changes. Organizations are encouraged to participate in the CMMI comment and improvement process.✍

Note

1. See Sandy Shrum's article on page 6 for more on staged vs. continuous representations of the CMMI.

References

1. *SW-CMM v2 draft C Software Engineering Institute. Software CMM, Version 2c*, Oct. 22, 1997.
2. *EIA/IS-731, Systems Engineering Capability Model*. Represents a merger of legacy systems engineering models under the auspices of the INCOSE and Electronic Industries Association. Available at www.incose.org
3. *IPD CMM Version 0.98*. Not publicly available.

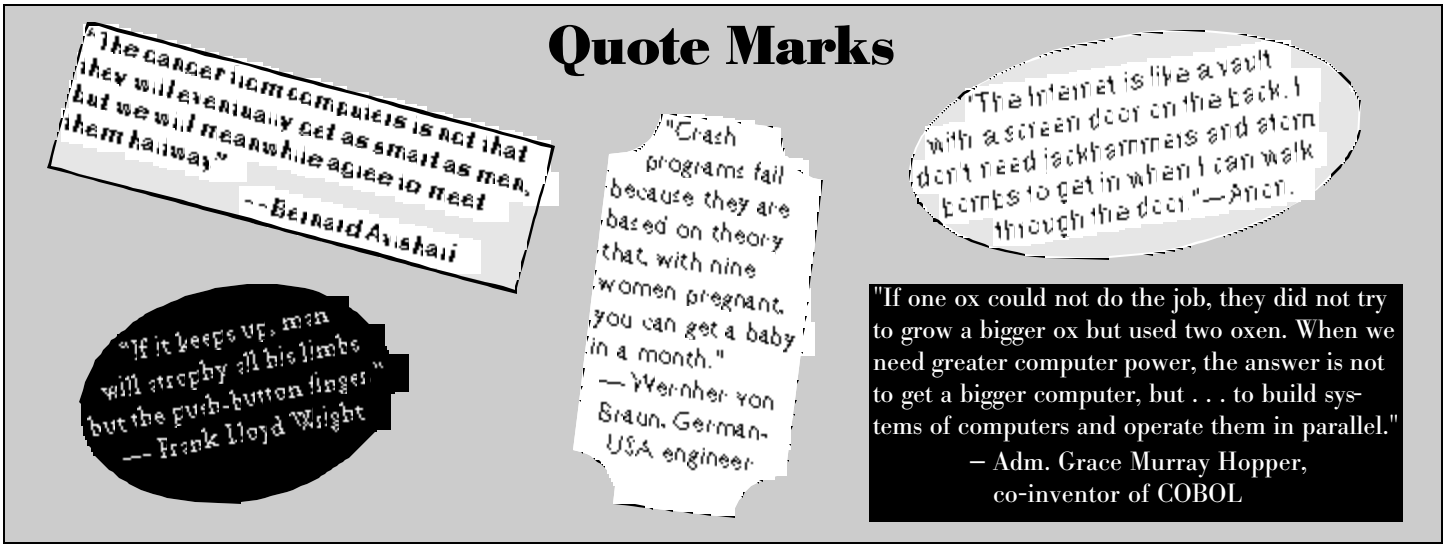
About the Authors

Aaron Clouse is a senior Principal Systems Engineer at Raytheon Systems Co. He has worked on several programs involving radar, sonar, global positioning systems, and cellular telephones. He is assigned to work engineering metrics and process improvement programs and is an author of CMMI.

P.O. Box 801, MS 8078
McKinney, Texas 75070
Voice: 972-952-2110
Fax: 972-952-2532
E-mail: Eajc@raytheon.com

Curt Wells consults on process improvement and assessment and teaches Process Oriented Systems Engineering for the University of Texas. He co-authored EIA/IS-731 and the Systems Engineering Capability Maturity Model, and is an author of the CMMI.

i-metrics
90 S. Camino Real
Uhland, Texas 78640
Voice: 512-76-9688
Fax: 773-913-0473
E-mail: cwells@i-metrics.net



Is CMMI Ready for Prime Time?

by Bill Pierce
Northern Utah Process Improvement Technologies

The impending release of the Capability Maturity Model-Integrated-Systems/Software Engineering [1] (CMMI-SE/SW, hereafter referred to as CMMI) has led to a lot of angst in the industry. Many organizations are asking themselves, and their customers, if they should adopt the CMMI as the framework for their process improvement efforts. Such a decision will have a great impact on their operations now and for a long time to come. CMMI is an improvement to existing models, but is it ready to take their place? This article is intended to help organizations decide whether to adopt CMMI based upon a review of the model and the experience of being part of the first pilot assessment using CMMI.

This article begins by describing the technique used for the pilot assessment, followed by discussion of some problems with the CMMI. These problems are described in terms of the difficulty of basing an organization's process improvement on the CMMI, and the difficulty of using the CMMI during an assessment. It is expected that the harder it is for an assessment team to use the CMMI, the less consistency there will be between assessments. Emphasizing the point that the CMMI is a step forward, some of the improvements made by the new model are discussed. Finally, an argument is made that most organizations should not move immediately to the CMMI.

The reader should note that the pilot assessment was conducted in December 1999, using CMMI SE/SW Version 0.2b. Training in the assessment methodology, the Standard CMMI Assessment Method for Process Improvement (SCAMPI) was provided immediately before the pilot assessment. All comments in this article are based upon those versions of the model and methodology, and do not acknowledge changes made since then.

Pilot Assessment

Two assessment teams participated in the pilot. They were both all-star teams in that there was a large proportion of experienced Lead and Candidate Lead Assessors, as well as representatives from industry and the CMMI Product Development Team (PDT). Observers from the PDT were at all team functions in order to learn as much as possible about how the model and methodology would be applied, but were not to intervene in the assessment process.

The two assessment teams reviewed all data independently and drafted separate findings. Both teams attended the same training on the new CMMI model and the new method of assessment, SCAMPI. Each team reviewed the organization's documentation, drafted questions for the interviews, and analyzed data. One team was designated as the A team; it questioned the interviewees and requested further documentation. The A-prime team could not ask questions, but attended all interviews and received copies of all documentation. The teams each drafted findings based on what they had seen and heard.

The findings from the two assessment teams were virtually identical. There was only one minor area of disagreement, and both teams stated that they discussed the area in depth, and could have decided either way. This verifies that the two assessment teams provided the same data would likely interpret the results in a similar fashion.

Process Improvement

The Capability Maturity Model for Software Version 1.1 [2] (CMM) was often criticized for being designed to operate in large organizations doing multimillion dollar projects. Small companies had to be convinced that it could be tailored and applied on small projects as well. The CMMI will surely receive even more criticism in this area, because many of its practices are based upon the experience of the Department of Defense and its contractors. The following process areas (PAs) all seem to have their roots in the large bidder/source selection environment: Supplier Agreement Management, Technical Solution, Verification, Validation, Product Integration, Data Management, Risk Management, and Decision Analysis and Resolution. It will be a real challenge to entice small, high-tech companies to adopt these kinds of practices into their daily business. Many will reject CMMI because it seems to insist upon a large bureaucracy to manage these activities.

Another significant issue is the sheer number of Process Areas and practices. There are 437 practices in the CMMI. An organization that is beginning to implement a process improvement program needs to focus on a small number of areas that will provide quick and measurable payback. CMM Version1.1 was criticized for being too large; the CMMI is even larger. Having eight process areas at Level 2 and 11 at Level 3 causes a fledgling organization to be daunted by the apparent magnitude of the effort required.

Process improvement according to the CMM for Software Version1.1 has earned a lot of buy-in from the industry. A major reason for the successful adoption of the CMM is that it is a framework for process, not a description of process. The practices intentionally describe *what* to do, not *how* to do it. The CMMI seems to take a step backward in that regard, and may not be as readily adopted as the CMM.

An example of the more prescriptive nature of CMMI is the Risk Management PA. In Risk Management, the first activity calls for identifying sources and categories of risk. The second has the organization define the parameters for categorization and for controlling the risk management efforts. The fourth requires assessing each risk for likelihood and consequence, and the fifth that every risk has a mitigation plan. These are all good things to do, but describe one particular method of handling risks. They are not necessary for all organizations, applications, or projects. Mitigation plans are normally developed only for high-impact risks; those that are either very likely or could be

catastrophic. This PA is clearly an example of a prescriptive model rather than a framework or guidance, and it will probably adversely impact the industry's willingness to adopt the CMMI. Organizations support the principle of risk management, but a model designed to facilitate process improvement should not prescribe one method over another. During the first pilot assessment, it was difficult for the interviewer to ask the interviewees appropriate, nonleading questions that would elicit answers about the particulars of how risks were managed. Again, this may be a very difficult task for assessment teams and could lead to inconsistent results.

Merged vs. Integrated

CMMI is less an integrated model than a merged one. There is much overlap between some Process Areas that serve to prolong the separation between the systems engineering and software engineering disciplines. It appears to an outsider as if the systems engineering part of the PDT developed their portions of the model, and the software engineers developed theirs, with insufficient cross-talk between the two groups. The different parts were then combined without enough concern for overlap. This "merged vs. integrated" appearance is strengthened by the tone of the activities; the Software PAs are written similarly to CMM 1.1, while the Systems PAs read more like Interim Standard 731. Much more work needs to be done to truly integrate the model.

Assessments using the CMMI

The CMM-Based Appraisals for Internal Process Improvement (CBA IPI) method of assessment allowed for an organization to rule a KPA as nonapplicable, without adversely impacting its maturity level. This was normally done only for Software Subcontract Management, since many development organizations do not contract out any work. The SCAMPI methodology did not allow for nonapplicable PAs during the pilot assessment. The organization made it clear that they did not subcontract any work, and so the assessment team did not rate the Supplier Agreement Management PA, ruling it out of scope. It is also likely that organizations will desire to place other PAs out of scope. Likely candidates are Data Management (not all organizations need a formal process and department to do this), Decision Analysis and Resolution, and possibly Technical Solution and Product Integration. The full scope of these last two may apply only in large organizations and projects. It is expected that this situation will occur on many assessments, and the SCAMPI methodology should clearly spell out what is to be done.

There are many practices in CMMI that are difficult for an assessment team to evaluate. This is especially true in the Systems Engineering Process Areas. Technical Solution has several of these practices. An example is Activity 4,[3] Develop design alternatives and selection criteria that consider the following:

- Life Cycle Cost.
- Technical Performance.
- Complexity.
- Robustness to product operations and the environment.
- Product expansion and growth.
- Cost drivers.
- Technology limitations.
- Sensitivity to construction methods and materials.
- Risk.
- Evolution of equipment drivers and technology.

Since the bulleted items are part of the activity, the organization must have implemented all of them. The laundry list of items in the Activity are all worthwhile tasks, but an assessment team must evaluate each of them and decide if the organization performs them as part of normal practice. There are only two ways to do this: Ask directly about them or ask about the technical solution process as a whole and hope they are all covered during the interviews. Our team tried both methods, and found that neither method works. If the interviewer asks the laundry list of items, the interviewee is overwhelmed by the sheer number of things and cannot possibly remember to answer them all, especially in the stressful environment of an assessment. There is not sufficient time to cover all the information required. If the respondents are not asked about the complete laundry list, they will certainly not cover all of the items and the organization will not receive credit for its practices. It is highly unlikely that a separate interview will corroborate the answers. These types of laundry lists should be covered the same way as in CMM 1.1; that is, as sub-bullets to the Activity. In that way, it is up to the assessment team to keep the laundry list in mind during the process, and focus on the issues that are deemed most important to the organization.

Decision Analysis and Resolution

Decision Analysis and Resolution (DAR) [4] is normally interpreted as a requirement for an organization to have a process to trigger a formal decision-making mechanism when a decision needs to be made on an issue of major concern to an organization's business. The PA goes on to describe some characteristics of both the triggering mechanism and the actual decision-making process.

It is very difficult for an assessment team to reach consensus on the activities in DAR. Our assessment team found that the interviewees had little knowledge of how these types of decisions were made, usually replying that they were not involved or it was not part of their jobs. The people who are involved in making the big decisions covered by DAR are rarely interviewed in an assessment.

The CEO or business unit executive would likely be a major player in such decisions, at least in small- or medium-sized organizations, and is typically the recipient of the assessment findings. That person is normally excluded from the data-gathering interview sessions and the Draft Findings feedback session. In order to build momentum for organizational change, the results of the assessment are presented first to the interviewees to gain their feedback and buy-in. Then, the Final Findings presentation is given to the senior manager in front of the entire organization. The SCAMPI method does not describe a method for interviewing the DAR-level decision makers and having them see the results for the first time as part of the final findings.

The goals described in DAR conflict with the true business

drivers in small and high-tech companies. The motivation for making decisions is often based upon the greater value of time-to-market needs rather than achieving the highest possible quality or productivity. Many of the small or high-tech companies have to be quick, nimble, and flexible, and cannot afford to follow a choreographed decision-making process that is triggered just when the decision is needed. Such a process inevitably delays the decision. Although it may be argued that such a process will usually generate a better solution, these companies' competitiveness is based upon rapid response to conditions. The accuracy of the decision is much less important than the speed.

Finally, the organization's decision makers will not be amenable to having the assessment team question them on their decision-making process. Many leaders of high-tech companies are likely to throw you out of their office if you ask them to describe how they make decisions! This puts the Lead Assessor and the assessment team members in an uncomfortable position. DAR, in its current format, will likely be handled differently on each assessment, and it will be difficult to get consistency.

As a final point on DAR, the new SCAMPI method does not allow this Process Area to be non-applicable during an assessment. It is likely that many organizations will rule DAR out of scope from both their process improvement program and their process assessments. This could result in DAR being an abandoned child of the CMMI, with very little consistent industry data or experience to validate its utility.

Institutionalization Common Features

Another major change to the structure of the model is that all of the institutionalization practices are grouped under an institutionalization goal. Each Process Area has a goal for institutionalization; the practices must be institutionalized as a managed process at Level 2, and as a defined process at higher levels. The change was probably intended to resolve a consistency problem with the CMM-Based Appraisal for Internal Process Improvement [5] (CBA IPI) method. In a CBA IPI, the assessment team needed to come to consensus that the organization demonstrated capability in at least one activity for each of the common features of every Key Process Area (KPA) [6]. Different assessment teams may get sufficient data on different practices but still find that the organization satisfies the KPA, or an assessment team may decide that a weakness on a particular activity prevents the organization from achieving that KPA. This method, however, did allow the assessment team to consider the organization's business needs in that decision.

In the SCAMPI, all of the institutionalization practices map to the last goal in each PA, and an organization must demonstrate capability in all of those practices to achieve that goal (and the PA). There are 10 institutionalization practices in each PA, resulting in 240 institutionalization practices the team must investigate and come to consensus on in a Level 5 assessment. Each of these practices must be performed as a defined process.

A defined process is a managed process that is tailored from the organization's set of standard processes. Deviations from the managed process are documented, justified, reviewed, and approved. A defined process also has clearly stated inputs, entry criteria, activities, roles, measures, verification steps, outputs, and exit criteria. A managed process is defined as being planned, documented, performed, monitored, and controlled at the local level.

There are 18 separate characteristics for a defined process mentioned in the above definitions. Therefore, there are 4,320 characteristics of process that an assessment team must verify just for the institutionalization practices! This is clearly an onerous requirement that an assessment team has insufficient time to accomplish, and will likely lead to less, not more, consistency between assessments.

The goal of investigating institutionalization is to ensure that the practices demonstrated for the assessment team are truly being used throughout the organization as a part of normal activities. In other words, they were not put into practice just for the assessment, either functionally (on the few projects being investigated) or temporarily (only shortly before the assessment). The assessment team must be confident that the practices will continue into the future after the assessment. This confidence can be achieved by listening closely for those instances where the organization demonstrates a lack of institutionalization, and following up with further questioning. A consistent pattern of successful implementation and continuous improvement of processes across the organization should be sufficient to enable the team to be confident that processes will continue into the future.

During our pilot assessment, it quickly became evident that the team could not verify all of the institutionalization practices across the organization. For example, practitioners must be trained in each of the managed and defined processes. Our team began investigating the training by asking people if they had been trained in each of the PAs as we discussed them. First, the repetition seemed to the team, and likely to the interviewees, that we were trying to trick them or catch them in a falsehood. Second, the repeated questioning on whether they have received training in all areas took an inordinate amount of time, especially compared to the value of the responses. Third, as discussed above, a laundry list approach is insufficient to get at all of the required training unless it becomes too leading to serve as corroboration. Fourth, verifying the training by reviewing records is inefficient and often fruitless. We wound up doing what most assessment teams do in CBA IPIs—sampling and investigating where we thought we had found inconsistencies. This approach should be standardized into the SCAMPI by describing it.

Improvements

Despite the problems described above, many improvements have been made. Requirements Management is now broken into two PAs, one at Level 2 and another at Level 3. It has also been explicitly described as being a management function throughout the development life cycle, rather than only being performed at the beginning of the project. The CMMI now defines eight different types of requirements, further emphasizing their importance to good project management across the life cycle.

In Project Planning, estimates must now be based upon the previous step (i.e., " . . .effort and cost . . . is determined by using historical data or models . . . from work product and task attributes"). Implementing Risk Management as a separate PA is

a great idea, and shows that the PDT was listening to industry concerns.

The CMM's awkward mapping of Key Practices to goals has been improved by having each practice map to a single goal (an exclusive many-to-one mapping for database designers). The mapping was evidently spelled out as one of the PDT's requirements, because it apparently has been applied without exception. Although this makes the assessment rating exercise much simpler, it imposes strict rules on the structure of the model that in some cases lead to a seemingly contrived or limited mapping of activities to goals.

Setting up Measurement and Analysis (M&A) as a separate PA is also an improvement. Each KPA in CMM 1.1 had an M&A Common Feature. Lower maturity organizations had difficulty understanding that the measurements were supposed to integrate across the entire process, allowing for coordinated management. The CMMI's separate PA for this should help to implement a more comprehensive metrics program. An argument could be made that Level 1 organizations cannot define proper metrics or design a system to collect and analyze them, so M&A might be a better fit at Level 3, but this is still a better approach.

Bringing systems engineering practices into the model should be a major advantage. Many software-only organizations were insistent that they did not do systems engineering, but all software must run on some computer system, and interface with others. Taking advantage of the rich history and experience of systems engineers can only help the software industry, and systems people can certainly benefit from the software experience, especially with process improvement.

Summary

The CMMI incorporates some major improvements over CMM 1.1. However, it also has some problems that need to be resolved. Organizations that have a mandate from customers to adopt CMMI should certainly do so, but if the goal is to achieve the benefit of process improvement, it may be better to wait for further model improvements. Major suppliers to DoD should be encouraged to implement as much as possible of the CMMI in the near future, but smaller companies will probably benefit most by waiting until many of these problems are fixed. In short, the CMMI is ready for release to a selected audience but probably not yet ready for prime time.✍

References

1. Ferguson, Jack and Ahern, Dennis, [et al.], *Capability Maturity Model-Integrated-Systems/Software Engineering,* Vol. II, Version 0.2b, September 1999.
2. Paulk, Mark [et al.], *The Capability Maturity Model, Guidelines for Improving the Software Process,* 1994.
3. Ferguson and Ahern, pp. 98-99.
4. Ferguson and Ahern, pp. 408-418.
5. Dunaway, Donna K. and Masters, Steve, *CMM-Based Appraisal for Internal Process Improvement (CBA IPI) Method Description,* CMU/SEI-96-TR-007, April 1996.
6. Software Engineering Institute, Carnegie-Mellon University, *CMM Based Appraisal: Team Training Participant's Guide,* 1998, slides F27 and F28.

About the Author

Bill Pierce has been involved in software process improvement for more than 15 years. He is authorized to lead CBA IPIs and to teach Introduction to the CMM. He has a master's degrees in business administration and engineering physics. He was a founding member of the SEPG in 1991 at Hill Air Force Base in Ogden, Utah. The SEPG started and managed an organizational process improvement program, attaining Level 3 in 1995 and Level 5 in 1998. The division was the first government organization and the sixth organization of any kind to achieve Level 5. Pierce was a leader in the process improvement efforts and managed one of the focus projects chosen for the assessment. In October 1999, Pierce founded Northern Utah Process Improvement Technologies. The company specializes in analyzing an organization's business needs and situation, designing and facilitating a reasonable and affordable process improvement program.

Northern Utah Process Improvement Technologies
1283 East 125 North
Ogden, Utah 84404-4142
Voice: 801-791-4621
E-mail: wdp@nupit.com
Internet: www.nupit.com

Coming Events

July 11-13
5th Annual Conference on Innovations and Technology in Computer Science Education
www.cs.helsinki.fi/events/iticse

July 16-18
7th IEEE Workshop on Computers in Power Electronics
www.conted.vt.edu/compel.htm

July 16-19
Congress on Evolutionary Computation
http://pcgipseca.cee.hw.ac.uk/cec2000

August 6-11
6th Annual International Conference on Mobile Computing and Networking
www.research.telcordia.com/mobicom2000

August 7-8
IEEE Workshop on Memory Technology Design and Testing
http://pcgipseca.cee.hw.ac.uk/cec2000

August 17-19
Designing Interactive Systems (DIS)

September 10-12
Collaborative Virtual Environments (CVE)

September 10-14
Very Large Databases (VLD)
www.acm.org/events has information on DIS, CVE, and VLD.

September 18-19
The Internet Challenge—The Utility Response to a .Com World
www.tdworld.com/marketing/interchall.htm

September 26-28
2nd Computer Security & Information Assurance Conference
www.certconf.org

October 15-19
Object Oriented Programming Systems Languages and Applications Conference (OOPSLA 2000)
www.acm.org/events

October 23-25
4th Symposium on Operating Systems Design and Implementation
www.usenix.org/events/osdi2000

October 30-31
3rd International Conference on Practical Aspects of Knowledge Management (PAKM 2000)
www.do.isst.fhg.de/workflow/events/index_e.html

April 29-May 3, 2001
Software Technology Conference 2001
www.stc-online.org

Letter to the Editor

In reference to George Jackelen's *The Need for a Useful Lessons Learned Database* [CROSSTALK, Jan. 2000], I always thought lessons learned were useful. Where I work, we have a mixed military and civilian workforce, and have at various times had committees, policies, or requirements to put together lessons learned. None have ever been successful. A few years ago, in a meeting with a newly arrived leader, I realized why.

In his briefing, the leader asked a question. My answer was that we had tried various approaches but our lesson learned was that the recommended approach was best for the circumstances. His answer was, "You have never tried it under my leadership." It became crystal clear to me why lessons learned never work. Everyone thinks he could have done a better job and would not have made that mistake.

Lloyd Pickering

CALL FOR ARTICLES

DECEMBER 2000
PROJECT MANAGEMENT
DEADLINE AUG. 1

JANUARY 2001
MODELING & SIMULATION
DEADLINE SEPT. 1

FEBRUARY 2001
CONFIGURATION MANAGEMENT
DEADLINE OCT. 2

MARCH 2001
MEASURES & METRICS
DEADLINE NOV. 1

APRIL 2001
REQUIREMENTS MANAGEMENT
DEADLINE DEC. 4

Please E-mail submissions to features@stsc1.hill.af.mil

Author Guidelines are available at www.stsc.hill.af.mil

Issues will not focus exclusively on one theme. We accept article submissions on all topics at all times.

Open Forum and BackTalk submissions welcome.

Please address any questions to Heather Winward at
801-775-5555 DSN 775-5555

Thank you in advance for your contributions!

w i crossed e s

In Keith Wegner's *Proven Techniques . . .* (June '00), the equation at the foot of page 27, " . . . derive the Kalman gain matrix from ? ?? $PH^T[HPH^T + R]$ for even six states . . ." should be ? ?? $PH^T[HPH^T + R]^{-1}$.

STC 2000 Photo Album

Photos clockwise from upper left

A pair of Bar J Wranglers fiddle away at the Old West Roundup.

The Salt Palace, site of STC 2000.

Birds-eye view of the exhibit hall.

James W. Moore signs a copy of his book for Lt. Col. Wayne Feltman.

Attendees queue up to register.

Errol Shim of Shim Enterprise Inc. confers with Larry W. Smith at the STSC booth.

Adm. Kenneth Slaght discusses the CMM with exhibitor Sylvia Centeno.

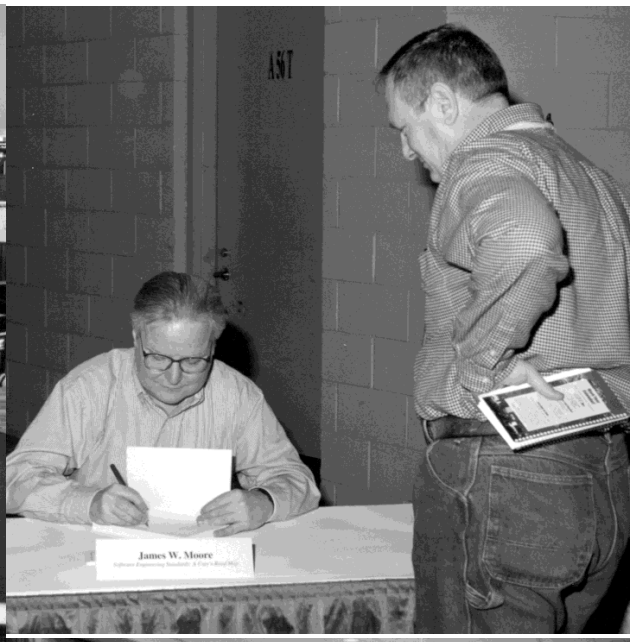
Col. Herbert B. Scherbinske (Ret.) congratulates Lt. Col. Joe Jarzombek on his retirement.

A magician enlivens a booth.

Deputy Under Secretary of Defense (S&T), Dr. Delores Etter, delivers a keynote address.

(Center Left) Informal afternoon social.

(Center Right) Dr. Roger Firestien delivers the Academic Keynote Address.



Photos by Randy Schreifels and Matt Welker

Requirements Elicitation in Open-Source Programs

By Lisa G.R. Henderson
Industrial Engineering Department, Missisippi State University

Requirements elicitation is an essential part of any software development activity and managing change to requirements once captured has proven to be an essential project management task. There is, however, a counterexample to this that exists today—the open-source community. This paper reviews the working of that community and suggests some accepted requirements of engineering practices that might be helpful.

Traditionally, requirements elicitation techniques have not been used in the open-source community. Part of the reason can be attributed to how the community works. To understand the possible role that requirements elicitation may have in open-source projects, an understanding of open source, definitions and components, must be achieved first.

What is Open Source?

The way an open-source program comes into being is that a developer writes a program for his own use to meet his own needs and then distributes it freely for the rest of the community to use if desired [1]. This type of program is not necessarily open source—it is just free software. There is more to an open-source program; the original programmer distributes the source code along with the binary version of the program and anyone willing to make an effort can check the code before compiling and running the program. Over the years, the term *open source* has meant different things. The Open Source Initiative (OSI) was created to try to standardize the definition and even license programs. According to the OSI definition, in order for software to be truly open source, it must exhibit certain characteristics, some of which are:

- Free redistribution.
- Inclusion of the source code.
- Allowance for modifications of the program/product distributed under the same terms as the original.
- Restricted distribution of modified source code to distribution of patches.
- No discrimination against individuals or groups.
- No discrimination against a specific field.
- Distribution of license.

The license attached to the product applies to all parts of the product and does not depend on a program being a particular software distribution

One of the key points listed above is that open-source software allows for modifications to be made to the source code of a program written by others. In *The Cathedral and the Bazaar*, Eric Raymond describes how an open-source program is created [2]. A programmer writes a (sometimes small) program and distributes it, usually by posting in a news group the location from which it can be downloaded. Others download it, use it, fix it (if warranted), and add features so that it is more useful. The creation is a group effort, and the program evolves (sometimes over a very short time). One of the central points of this paper is that the program users become its co-developers.

The Growth of the Open-Source Community

This kind of development has produced good and popular programs. Some are so popular that commercial companies have started to support them. Examples include Corel announcing its intention to port its entire WordPerfect Suite to Linux [3]. Other companies have decided to try making use of the open-source community to create products, like Mozilla [4]. These announcements and others have brought attention to the open-source community and to the programs that are available for free to the public. The public has responded by using them. For example, Linux, one of the most popular open-source operating systems, has seen its user base grow to more than 7 million people [5].

Netscape's Mozilla project cannot be considered the typical open-source project, since it failed to comply with one of the *rules* [2} of open-source projects by not first releasing a working product to the community for inspection and modification. Once the Web browser was successfully working, the project picked up speed. Netscape employees direct the Mozilla effort, but many more individuals work for free [4]. Netscape allows anyone to contribute to the project. Even nonprogrammers can contribute by helping test the product.

With the popularity of open-source programs growing, more and more new users will not be the same ones that have typically used open-source programs in the past [3]. Nor will they all be programmers. These new users will not become co-developers, except for testing, unless they want to learn how to program. However, they still want and need functionality, and need to be able to communicate that need to developers within the open-source community. Users can always join in discussions found in the pertinent newsgroups, but often discussions are so technical that they get lost or do not feel comfortable.

Bridging the Gap

The Free Software Bazaar was created [6] to help these new members of the open-source community reach the developers. It is a Web page where nonprogrammers can post monetary rewards for programs and where programmers can find projects that might be of interest. If someone wants to request a program, he posts his request and how much he is willing to pay to have the work done. A programmer gets in touch with the original poster to let him know that he is starting work. Like the Mozilla project, the projects listed on the Free Software Bazaar are not the typical open-source projects, since these projects are intended for customers other than the developer.

For the typical open-source project, requirements elicitation is not needed. A listing of requirements is usually made, but there is no need to go through most of the techniques normally used to ensure that the developer understands what the

customer wants (the customer and developer are the same person). But that is not true for the Mozilla project or projects posted on the Free Software Bazaar.

One thing the Mozilla project does not do well is elicit requirements from its customers. If a great deal of time is spent searching its Web site, there are instructions informing customers how to get their requirements added to the *wish list* of features for the web browser [4]. Most people, however, would give up long before finding these directions. Once found, they are directed to a news group frequented by Netscape employees.

The Free Software Bazaar also does not do well when obtaining requirements from customers [6]. It requests that customers' requirements be unambiguous, understood, and complete. Then it states that it cannot be changed while the software is being developed.

Having complete and unambiguous requirements is fundamental in the development of any software project. It should be a central issue for projects promoted on the Free Software Bazaar Web site, since the people involved in the projects listed change over time. Just because a customer posts a project and a developer signs up to work on it does not mean that there will be no one else involved. Others who are also interested in using the finished program can post their monetary contribution as well. Sometimes this additional funding is necessary to get a programmer interested in the job, but it results in having two or more different customers. Sometimes other programmers might also be interested in the project and volunteer their help. They contact the original poster who puts them in touch with the person who is working on the project. As the project progresses, more and more people can be added to the project on the customer side and on the developer side. This process sounds like a developer's nightmare unless the rules on the bazaar page are brought to mind. One rule is that the requirements are to be stated unambiguously and, once stated, cannot be changed.

Requirements Elicitation

Stating requirements without ambiguity is not easy, however, as most developers know. There are many techniques used in eliciting requirements, but which ones fit into the open-source community? With a typical software development project, the developer meets with the customer at the beginning of a project, but that simply is not practical in a global open-source community. The costs of traveling the necessary distances are prohibitive—especially when considering that the customers and developers could be in many different countries, and the cost of these projects are usually in the $20 to $2,000 price range. As meeting face-to-face is not practical, the requirements elicitation process should be conducted over the Internet.

Joseph Goguen and Charlotte Linde have listed several typical software requirements elicitation techniques in their paper [7]. They are:

- Introspection.
- Questionnaire interviews.
- Open-ended interviews.
- Focus, application development groups.
- Discussion.
- Protocol analysis.
- Discourse analysis

Many of these are easily ported to the open-source community.

Introspection

No matter what kind of software project, this technique has to be used. The developer cannot understand requirements without thought and imagination. There is, however, nothing about introspection that guarantees the way the developer understands the requirements is the same as the way the customer understands them. Other techniques are also needed.

Questionnaire Interviews

This type of interview is ideal for the open-source community. It can be easily implemented within a Web page and has the added advantage of being something all possible open-source customers are used to seeing. It still has the drawbacks the technique is known for, namely that the possible choices may not reflect the real response the customer wants to give [7].

Open-Ended Interviews

This type of interview is also seen a great deal on the Internet. It would require nothing more than a Web page with forms that the customer completes. It falls prey to the *say-do* problem where people know what they need the system to do, but do not know how to describe the process to someone else [7].

The two types of interviews could be combined with each survey question having a list of possible answers and a text box at the end if they feel that all of the listed choices are inappropriate. If the e-mail address of the person completing the survey is required, then the developer can send a message asking about any responses he does not understand.

Focus and Application Development Groups

This kind of group interview can be easily accomplished over the Internet by using a forum or a chat room. Lag can be an annoyance, but most people who surf the Web are familiar with this phenomenon. Another more serious problem with using a forum or a chat room is that all of the interested parties are not necessarily in the same part of the world. This makes finding a time when all parties can *meet* quite difficult. A focus/development group might be better implemented using a bulletin board, so the developer can post the interview questions, give everyone a chance to respond, and continue the question/answer session over a longer period of time.

Discussion

This technique is widely used by the open-source community—not for eliciting requirements, but for communicating program fixes, giving help with a project, or simply discussing the everyday problems with a program or job. It is typically implemented through newsgroups or mailing lists, and an applicable one can almost always be found no matter what the project.

Protocol Analysis

Verbal analysis involves recording someone's actions while he explains what he is doing and why he is doing it in an effort to understand exactly what he needs the system under development to accomplish [7]. This type of technique simply is not applicable because the projects are not that complex or that expensive—yet. It could easily be implemented over the Internet if the occasion arose. The recorded information would need to be made available to the developer through e-mail, ftp, or even the postal service.

Discourse Analysis

One advantage of having a *conversation* over the Internet, whether in chat rooms, forums, bulletin boards, or newsgroups, is that everyone has an equal opportunity to present their views. No one can interrupt another's sentence or story. Whenever someone has something to say, he or she enters it into the conversation without having to wait for someone else to finish. Taking turns is not very important in this medium for the same reason. It is not an environment conducive to someone talking, someone else responding, followed by someone else. These advantages are disadvantages with discourse analysis, which relies heavily on these things [7].

Disadvantages of Using the Net

There are disadvantages to using the Internet as well. No one can see another person; therefore, they miss visual cues such as body language and facial expressions. Tone of voice is absent. Someone dropping out of the conversation may go unnoticed, and it is easy to misconstrue or misunderstand what another individual's input means.

Reaching a Consensus

With the near certainty of having multiple customers when all the requirements have been gathered using the above techniques, everyone has to agree on them. It is quite possible that one person would need a feature, while another is opposed to including it. Customers and developers need to reach a consensus.

The Delphi technique was developed to do just that—achieve a consensus among a group of people [8]. The developer lists the requirements and asks for opinions from all of the customers. These opinions are listed anonymously and sent to each customer. They make comments about all of them, and send them back to the developer. After a few rounds of this, a group generally reaches a consensus. This technique is easily implemented over the Internet through e-mail, but could just as easily be been done with a slightly altered bulletin—one that would not post the name of the poster and therefore achieve the anonymity missing from a regular bulletin board.

The CONOPS Document

Another tool is the concept of operations (CONOPS) document [9]. In his paper, Richard Fairley writes that the customer should prepare the document for maximum effectiveness. With the possibility of having many different customers in many different parts of the world, most of which have never heard of the document, it would be better for the developer to create the document using any of the above techniques to clarify ambiguous requirements. Using the CONOPS document (or a modified version) has the added advantage in that it can also be used to advertise the project—drawing in more paying customers and developers. Once the project is finished, it can still be used to increase the number of users, who can later possibly become co-developers.

Once all parties agree upon the requirements, maybe formalizing the agreement by having everyone sign off on the CONOPS document, it is not unreasonable to freeze requirements and forbid changes as the Free Software Bazaar currently dictates. The reason that it is not unreasonable is that all open-source programs evolve, and developing the software is only the first step. Once the first version is complete and has been delivered, it will join the ranks of all other open-source software that people use, fix, and modify to meet their needs.

Conclusion

Requirements elicitation is a necessary part of all software projects when there is a possible misunderstanding between the customer and the developer. The open-source community has never used any of these in the past, but is rapidly approaching the time when they will be essential. Almost all of the techniques used in a more typical software development project can be applied within the community by using the tools available on the Internet. The CONOPS document in particular should be a boon to developers as a way for them to promote their software, in addition to its value as a contract with the customer. Getting the requirements correct, with the help of requirements elicitation techniques, and creating better software in the first step of the evolution of the project can only enhance the image the rest of the world has of the open-source community and the software it creates. ✍

References

1. Perens, Bruce and Raymond, Eric *The Open Source Page*, Feb. 24, 1998. Available at www.opensource.org
2. Raymond, Eric, *The Cathedral and the Bazaar*, Feb. 10, 1998. Available at www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html
3. Sykes, Rebecca. Linux, Linux Everywhere, *PC World News*, Nov. 18, 1999. Available at www.pcworld.com/pcwtoday/article/0,1510,13908,00.html
4. Endico, Dawn, on www.mozilla.org Sept. 13, 1999.
5. McHugh, Josh, For the Love of Hacking, *Forbes*, Aug. 10, 1998. Available at www.forbes.com/forbes/98/0810/6203094a.htm
6. Boldt, Axel, *The Free Software Bazaar*, Aug. 4, 1999. Available at http://visar.csustan.edu/bazaar
7. Goguen, Joseph A. and Linde, Charlotte Techniques for Requirements Elicitation, In *Software Requirements Engineering, 2nd Ed.*, edited by Richard H. Thayer and Merlin Dorfman, IEEE Computer Society Press, Los Alamitos, Calif. 1993.
8. Stahl, Nancy N. and Robert J. Stahl, We Can Agree After All! Achieving Consensus for a Critical Thinking Component of a Gifted Program using the Delphi Technique, *Roeper Review*, Dec. 1991.
9. Fairley, Richard, The Concept of Operations: The Bridge from Operational Requirements to Technical Specifications, In *Software Requirements Engineering, 2nd Ed.*, edited by Richard H. Thayer and Merlin Dorfman, IEEE Computer Society Press, Los Alamitos, Calif. 1996.

About the Author

Lisa Henderson has been a member of the engineering graphics group of the Industrial Engineering department at Mississippi State University, where she has taught for the past 12 years. She has a bachelor's degree in chemical engineering, and has worked toward master's degrees in chemistry and computer science.

Post Office Box 9542
Mississippi State, Miss. 39762
Voice: 662-325-7217
Fax: 662-325-7618
E-mail: lgr1@ra.msstate.edu

Give Us Your Information, Get a Subscription

Fill out and send us this form for a free subscription.

OO-ALC/TISE
7278 FOURTH STREET
HILL AFB, UTAH 84056-5205
ATTN: HEATHER WINWARD
FAX: 801-777-8069 DSN: 777-8069
VOICE: 801-775-5555 DSN: 775-5555

Or use our online request form at www.stsc.hill.af.mil

FULL NAME:_____

RANK OR GRADE:_____

POSITION OR TITLE:_____

ORGANIZATION OR COMPANY:_____

ADDRESS:_____

_____

BASE OR CITY:_____ STATE:_____

ZIP:_____

VOICE: COMMERCIAL_____

DSN_____

FAX: COMMERCIAL_____

DSN_____

E-MAIL: _____@_____

BACK ISSUES MAY BE AVAILABLE

(PLEASE INDICATE THE MONTH(S) DESIRED.)

MARCH 1999_____CONFIGURATION MGMT.

APRIL 1999_____SQA

MAY 1999_____CMM LEVEL 5

JUNE 1999_____MEASURES AND METRICS

AUGUST 1999_____SOFTWARE ACQUISITION

SEPTEMBER 1999_____D II COE

OCTOBER 1999_____BEST PRACTICES

NOVEMBER 1999_____CHANGE MANAGEMENT

DECEMBER 1999_____SOFTWARE EVOLUTION

FEBRUARY 2000_____RISK MANAGEMENT

MARCH 2000_____EDUCATION & TRAINING

APRIL 2000_____COST ESTIMATION

MAY 2000_____THE F-22

JUNE 2000_____PSP/TSP

Gilligan's Integration

To be sung to the tune of the Gilligan's Island theme . . .

Just sit right back
And you'll hear a tale
A tale of a noble quest
That started from a cubicle
A model now to test.

The lead was a strapping Gov'mt guy
An Assessor brave and bright
The team members embarked that day
Level 5 in their sights
Level 5 in their sights

The queries started getting rough
The KPAs were tossed
If not for the courage of the fearless crew
The Assessment would be lost
The Assessment would be lost

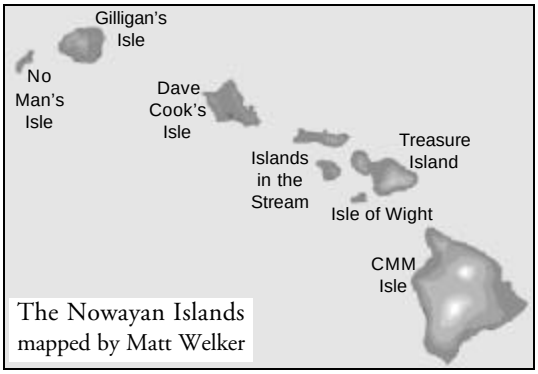
A framework was identified
Continuous and staged
With new PAs, Change Agents, too
The Engineers to engage
And Management
With Schedule, Cost, Quality
Here on CMM Isle!

Now this is the tale of our software team
They're here for a long, long time.
They'll have to make the best of things,
It's an uphill climb.
With management and teamwork, too
They'll do their very best
To make process repeatable
With fervor, zeal, and zest.

No fuss, no muss, no heroes here
Don't forget the TSP
Like Mr. Watts S. Humphrey
It's progressive as can be.

Now you can be the judge my friends
If this is worth your while
Before you launch your S-P-I
Don't be an imbecile!

Heather Winward (Mary Ann), Shim Enterprise Inc.



The Nowayan Islands
mapped by Matt Welker

Providing

Independent

Program

and

Capability

Assessments

Sponsored by the Computer Resources Support Improvement Program (CRSIP)

CrossTalk
Ogden ALC/TISE
7278 Fourth Street
Hill AFB, UT 84056-5205

BULK RATE
US POSTAGE PAID
Permit No. 481
Lenexa, KS